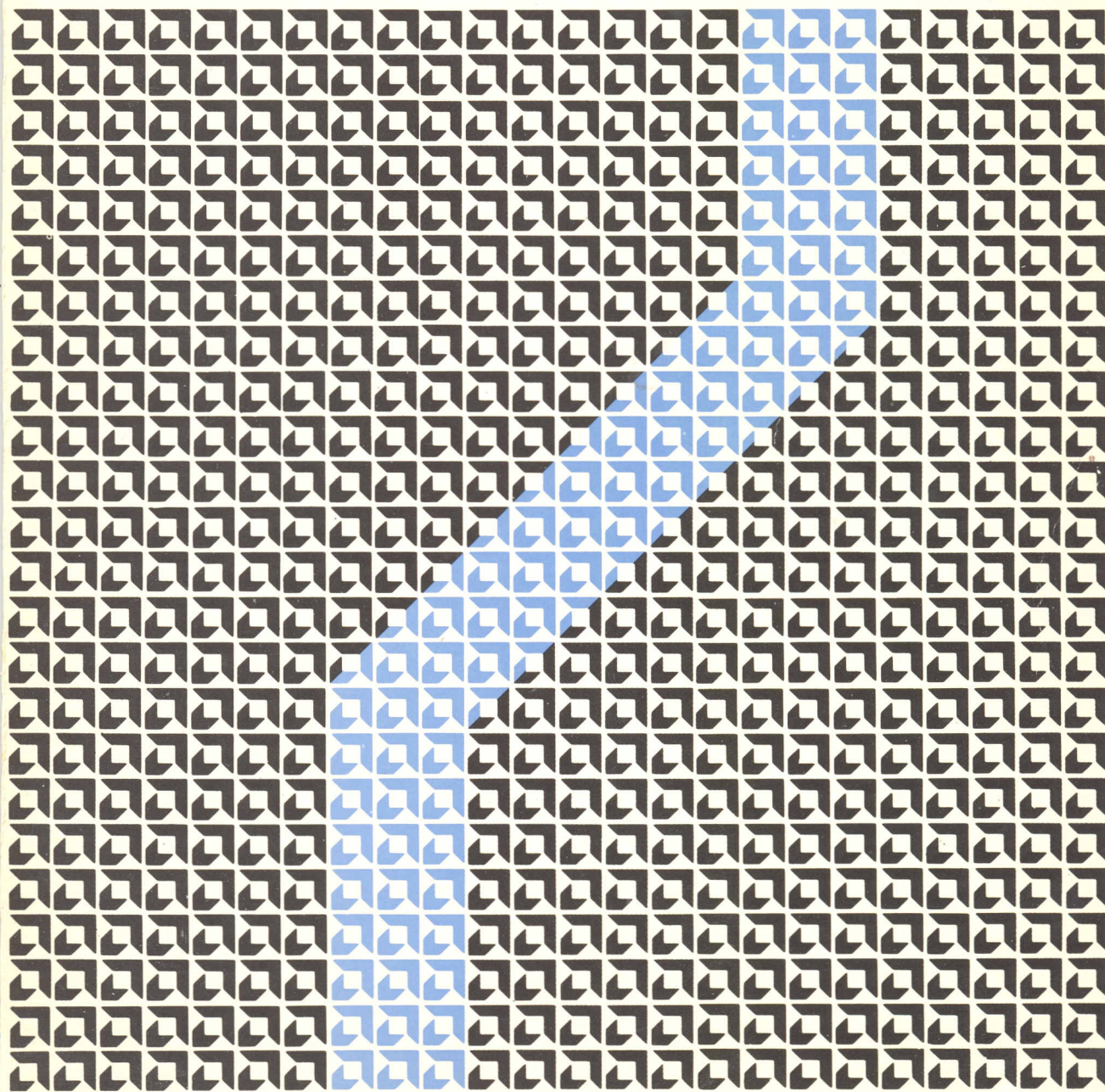
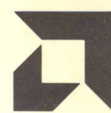


Advanced
Micro
Devices

Am9500
Peripheral Products
Interface Guide





Advanced Micro Devices

Am9500 Family Interface Manual

Copyright © 1980 by Advanced Micro Devices, Inc.

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics. The company assumes no responsibility for the use of any circuits described herein.

901 Thompson Place, P.O. Box 453, Sunnyvale, California 94086
(408) 732-2400 TWX: 910-339-9280 TELEX: 34-6306

Printed in U.S.A. 11-80 RMC-10

TABLE OF CONTENTS

INTRODUCTION

1. CHIP SELECT – DEVICE ENABLING	
Wait State Generation	1-3
2. INTERFACE CIRCUITS	
Am9511A	2-1
Am9512	2-5
Am9513	2-9
Am9517A	2-13
Am9518	2-18
Am9519A	2-20
3. SUMMARY	
Processor/Peripheral Selection Chart	3-1
4. Am9500 FAMILY DATA SHEETS	
Am9511A	4-1
Am9512	4-27
Am9513	4-47
Am9517A	4-73
Am9518	4-88
Am9519A	4-90
5. MICROPROCESSORS	
AmZ8002	5-1
Am8085A/Am8085A-2/Am9085ADM	5-22
APPENDIX	
Commitment to Excellence	A-1

INTRODUCTION

In today's world of increasing demands placed on microcomputer-based system applications, the "smart," high-performance peripheral can offer significant advantages. These products can, in many cases, expand the application and extend the life cycle of the system without the extensive hardware and software changes required by a full system redesign.

Advanced Micro Devices recognized these system needs and in 1978 began introduction of a second generation family of high-performance intelligent peripheral products designed for use with all general-purpose 8-bit CPUs and our AmZ8000 Family of 16-bit machines. This manual describes various interfaces between the AmZ8000, Am8085A, Z80 and 6800 microprocessors and AMD's 9500 Family of proprietary peripheral products.

This manual also presumes a basic familiarity with the microprocessors and peripheral circuits mentioned. We suggest that data sheets be handy for quick reference when referring to the various interface circuits.

Chip Select — Device Enabling

In order for the CPU or host system to access and communicate with other peripherals and support devices, some method must be used to select and manipulate these devices in a controlled manner. The CPU or host can control the peripheral and support devices through two types of conventions: memory-mapped or input/output (I/O).

Essentially, the memory-mapped technique is an I/O handling procedure where the peripherals are accessed as memory locations; it assigns areas of memory address space as I/O addresses. This architecture allows the CPU to manipulate devices by using the same instructions used to handle memory operations. Referring to Figure 1, note that both the peripherals and memory share the same common address and data busses. Each peripheral and memory location has a unique address which distinctively selects a device.

Figure 2 shows the second method which treats the peripheral device as an I/O peripheral. This method isolates peripheral devices from memory devices, but provides a direct means of communications with the CPU. In most general-purpose CPU systems, however, the flexibility of the software is limited by the

type of instructions available to manipulate I/O devices. The I/O instructions usually available are IN and OUT, which deal directly with the CPU's accumulator only. Thus, this technique becomes less effective due to the constraints of the simple I/O instructions.

Both conventions require some form of selecting one or more devices depending on the CPU requirements. Four different selecting techniques could be used, they are random logic, decoders, comparators and mapping PROMs. Since the random logic technique using SSI, gates, inverters, etc., can be implemented more easily than the other techniques, it will not be discussed.

As for decoders, AMD manufactures many different types of decoding devices, including the following:

Am25LS138	1-of-8 Decoder
Am25LS139	Dual 1-of-4 Decoder
Am25LS2438	1-of-8 Decoder with Three-State Output
Am25LS2536	1-of-8 Decoder with Control Storage
Am25LS2537	1-of-10 Decoder with Three-State Output
Am25LS2539	Dual 1-of-4 Decoder with Three-State Output

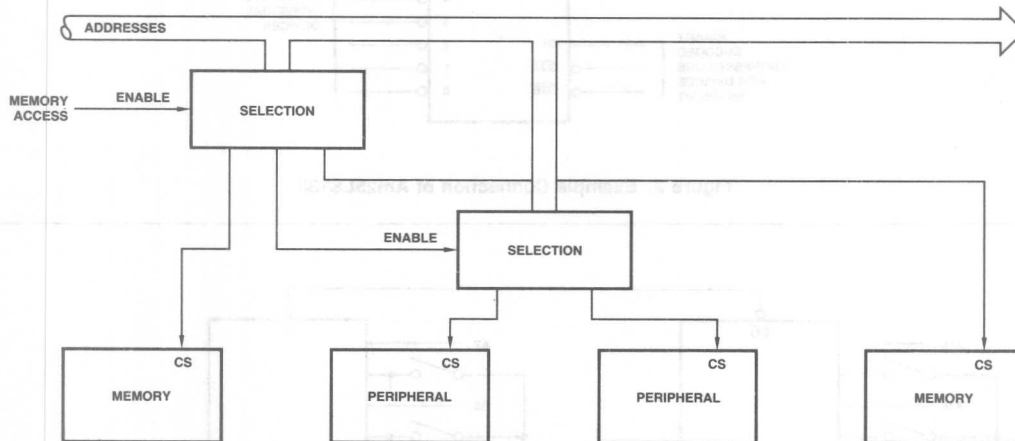


Figure 1. Memory Mapped Technique

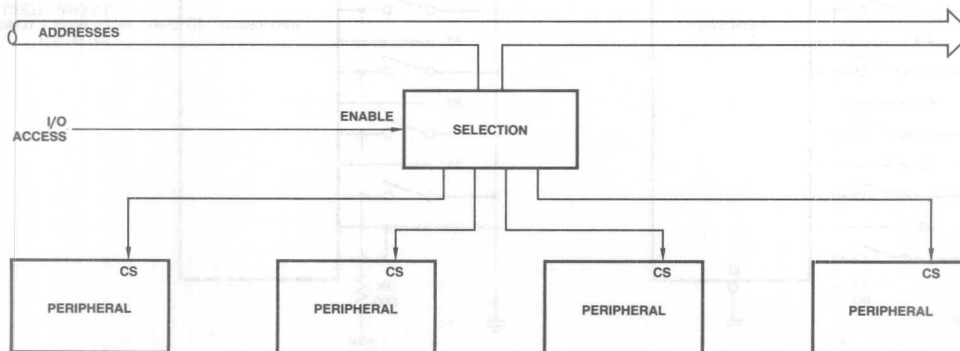


Figure 2. I/O Technique

Chip Select – Device Enabling

Decoder logic is used to create individual gated signals which selectively enable one device. The decoder normally connects to the address bus with its inputs tied to the proper individual address bit lines for the required address range.

Figure 3 uses an Am25LS138 to show the decoding of the lower bit addresses.

Note that the G1, G2A and G2B inputs determine where in the address space these eight device locations will reside. Assume, in the example, that G2A and G2B are connected to address bits A₃ and A₄, while A₅ through A₁₅ are NOR'd together and tied to G1. This configuration would then insure that the devices are decoded for the lowest eight address locations. Besides selecting devices, decoders could also be used to decode control signals as seen in AmZ8002 to Am9511A interface example on page 2-1.

Comparators offer another method to enable devices by providing an "equal to" condition when two patterns are compared. Depending on the type of comparator, 4- or 8-bit words can be compared to obtain a "greater than," "less than" or "equal to" logic expression. Figure 4 shows two Am25LS2521 8-bit comparators handling a 16-bit address range. By selecting the switch positions, a device can be enabled anywhere within a 65K addressing range.

The last technique mentioned makes use of the mapping PROM to control the device address to memory location relationship. Mapping PROMs translate (map) sequential addresses into different independent memory locations as opposed to the logical binary pattern restriction of decoders. In fact, any device can be accessed from any memory location by simply changing the PROM program.

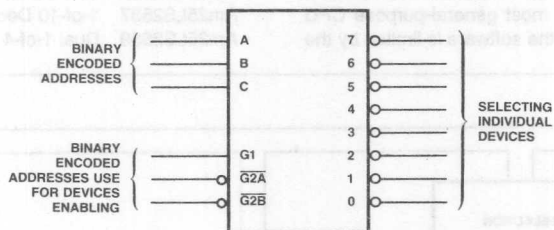


Figure 3. Example Connection of Am25LS138

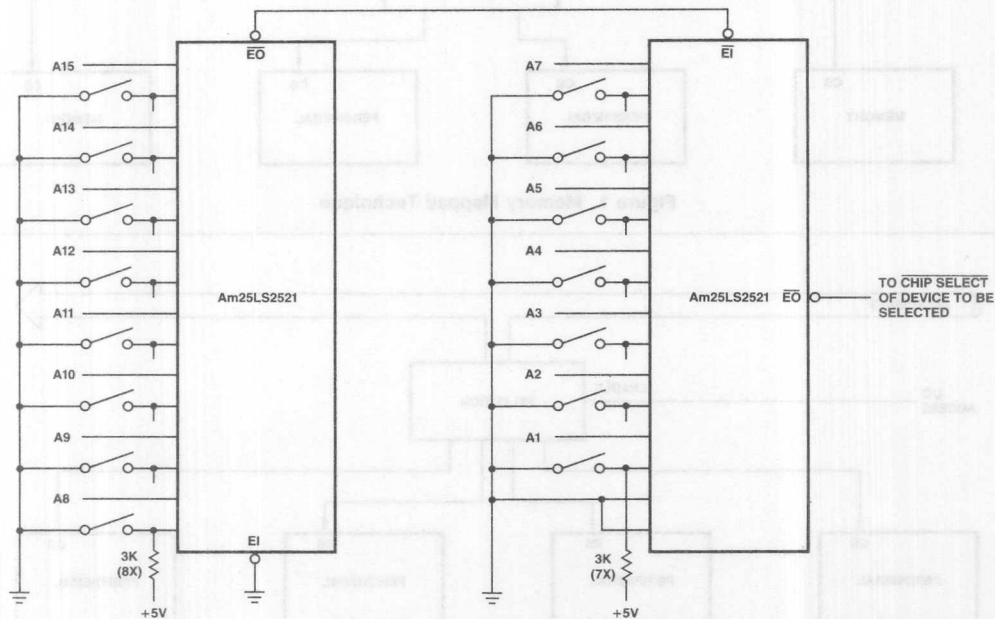


Figure 4. Device Enable Using Comparators

Figure 5 shows an Am27S13 512 x 4-bit PROM that can enable four separate devices within a 512 address range. Within this range, any of the four devices could be enabled in any random combination.

One of the address inputs of the PROM could be used to force a condition known as "booting" which initializes the system to a known operating state. For example, a known device could contain the starting address of a system monitor. The device could even retain the monitor in a ROM device and recall it only when the device is addressed. However, as in the example shown above, this reduces the device address range from 512 to 256 locations.

WAIT GENERATION

Wait state generation is performed on microprocessors to retard the speed of the microprocessor so that slower memories and peripherals could be used. In the wait state, the microprocessor enters into an idle loop or a complete halt condition. Thereafter,

the microprocessor cannot resume normal operation until the control signal that caused the wait state is negated. Usually this control signal is called the "READY" line.

There are different methods to generate the wait state. One method uses a resettable monostable (one-shot) multivibrator. Figure 6, shows an Am26S02 one-shot used to produce a wait signal. The duration of the wait state is determined by the RC constant. The one-shot would be triggered when the Chip Select (\overline{CS}) line of the peripheral is selected (goes LOW). This causes the READY line to go LOW and the processor enters a wait state for the specified period. Some microprocessors may require the system clock to be halted in a known state to perform this wait state operation.

A more digital synchronous method can be performed by using two positive D-type flip-flops as shown in Figure 7. The clock input of the first flip-flop connects to a signal which contains unique information that indicates the start of a new machine cycle. The second flip-flop clock is tied to the system clock.

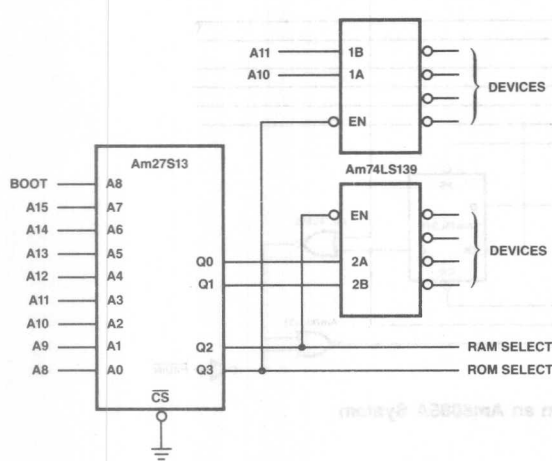


Figure 5. Mapping PROM Configuration

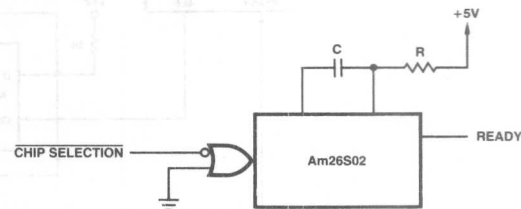


Figure 6. Wait State Generated by One-Shot

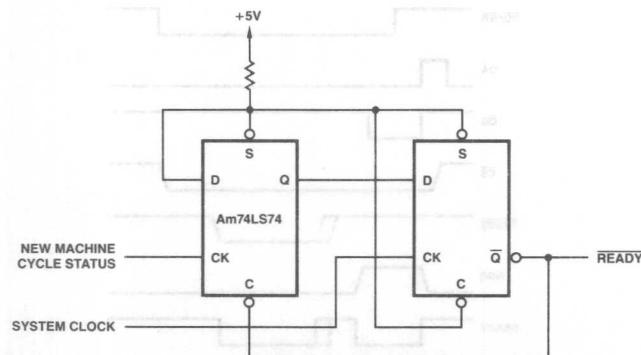


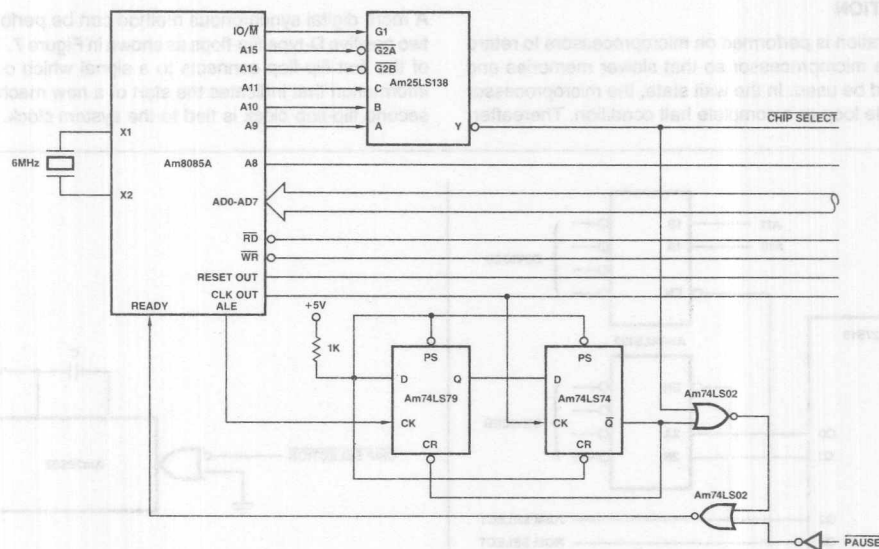
Figure 7. Wait State Generated by Two Flip-Flops

Chip Select – Device Enabling

A better explanation of its operation uses the Am8085A example shown in Figure 8. The Am8085A samples the READY line at the rising edge of T_2 to determine if any wait states are required. The \overline{RD} and \overline{WR} signals become true after address lines are stable. The \overline{RD} to \overline{PAUSE} delay is 150nsec max. To extend these control signals to Am9511A, the external logic shown must insert an initial wait state, because \overline{PAUSE} may not become true early enough (at the T_2 state rising clock) to meet the requirements of the Am8085A.

To insert a wait state when the device is accessed, the $\overline{\text{PAUSE}}$ output is inverted and fed into one input of NOR gate 1. When

Address Line Enable (ALE) goes HIGH, the Q output of D flip-flop A goes HIGH. The \bar{Q} output of the D flip-flop B goes LOW at the rising edge of clock in state T_1 and clears A. The READY goes LOW when \bar{Q} of B goes LOW, early enough to be sampled at T_2 rising edge. At the next rising edge of the clock in the T_2 state the “not Ready” condition is sensed by the processor and the \bar{Q} output of flip-flop B goes HIGH. PAUSE may become true sometime after \bar{Q} of B goes HIGH and will thereafter introduce further wait states. The READY will be deactivated when PAUSE becomes inactive (HIGH) thus stopping the insertion of wait states.



a) Example Wait Generation in an Am8085A System

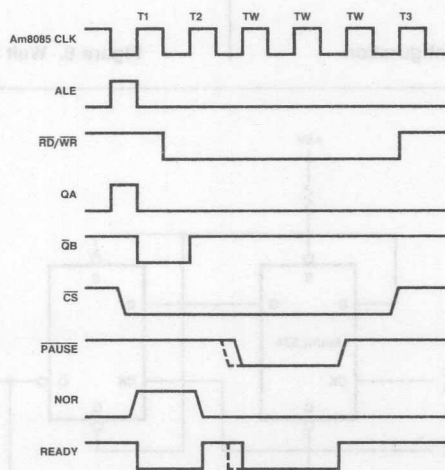
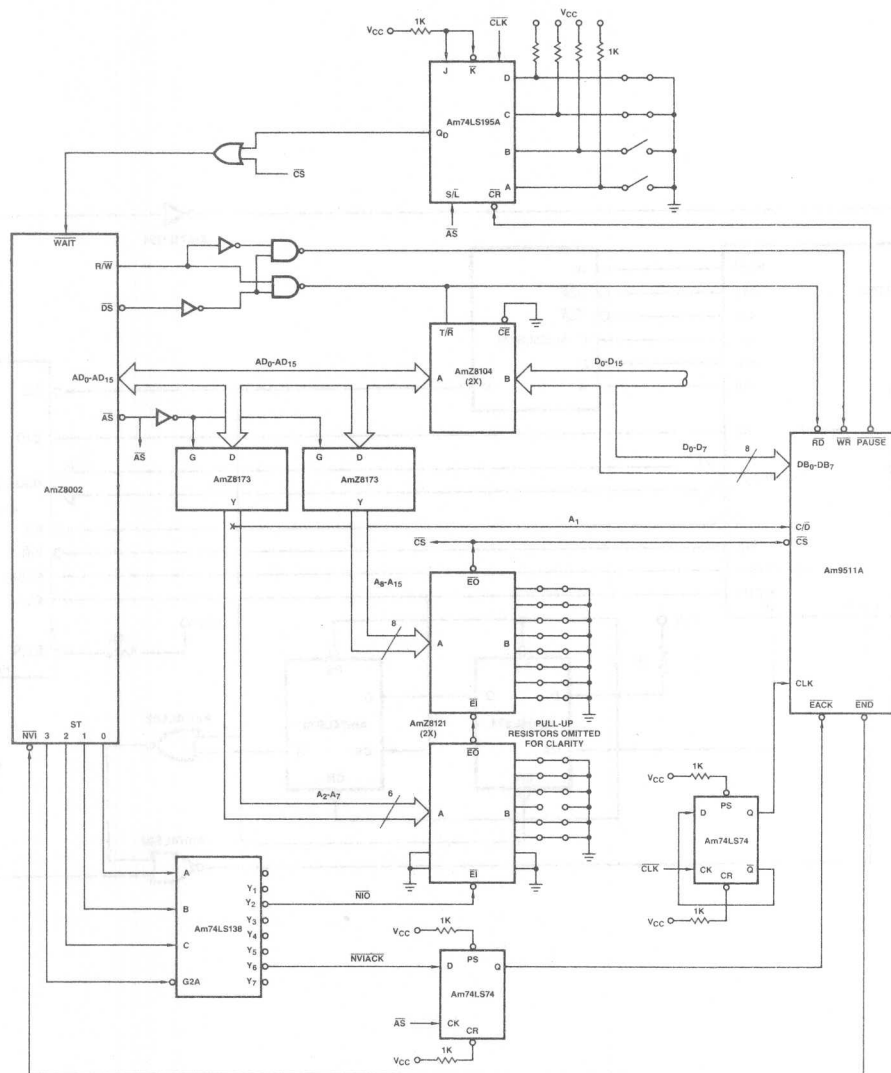
**b) Timing Diagram**

Figure 8.

Interface Circuits

Am9511A to AmZ8002 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

In this interface, the 16-bit AmZ8002 accesses the Am9511A in the byte mode during I/O transactions. The Am9511A data bus may be connected to either the upper or lower eight bits of the AmZ8002 bus (it is connected to the lower bits in this diagram). Therefore, I/O address should be odd.

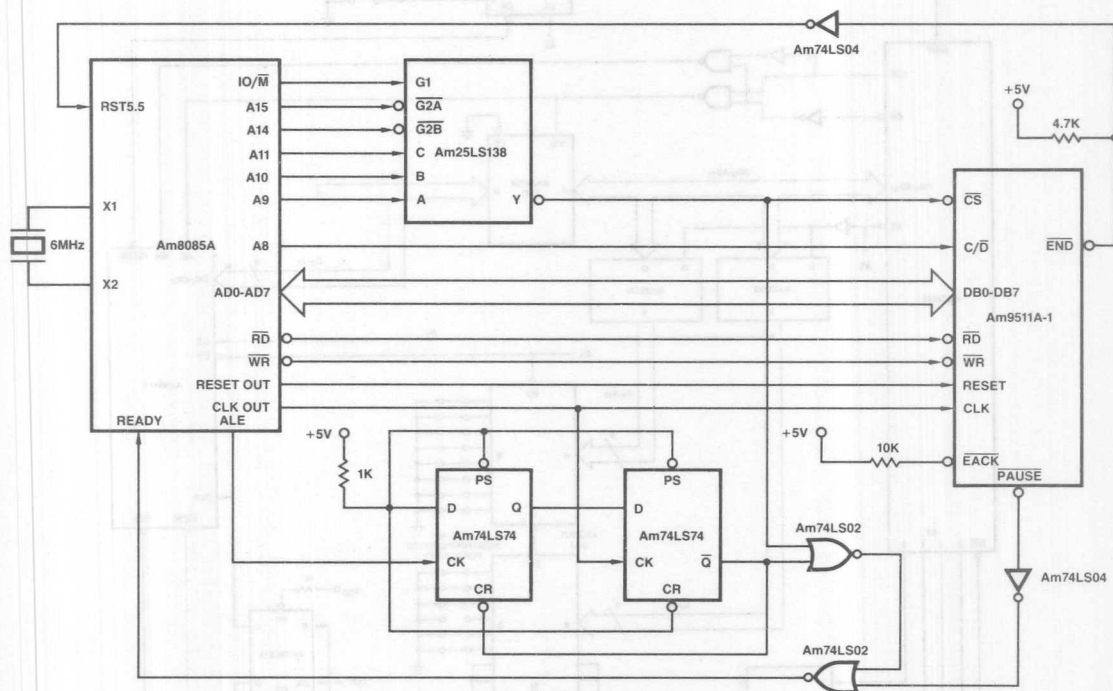
Since the read access time of the Am9511A is relatively long, the AmZ8002 must wait during each read access. The **PAUSE** output, however, cannot be tied directly to the wait input (**PAUSE** would arrive too late to wait state AmZ8002). The Am25LS195A

4-bit shift register is used to issue a wait state to the CPU immediately at address strobe if chip select to the Am9511A is present. The **Q_D** output will remain LOW for two clock periods. If **PAUSE** is LOW during this period, the **WAIT** line will remain LOW. After the **PAUSE** line returns HIGH, the **Q_D** output will go HIGH after two clock periods.

The circuit diagram shows an interrupt driven configuration. The **END** signal goes LOW and activates the Non-Vectored Interrupt **NVI** at the end of an Am9511A operation.

Am9511A-1 to Am8085A Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

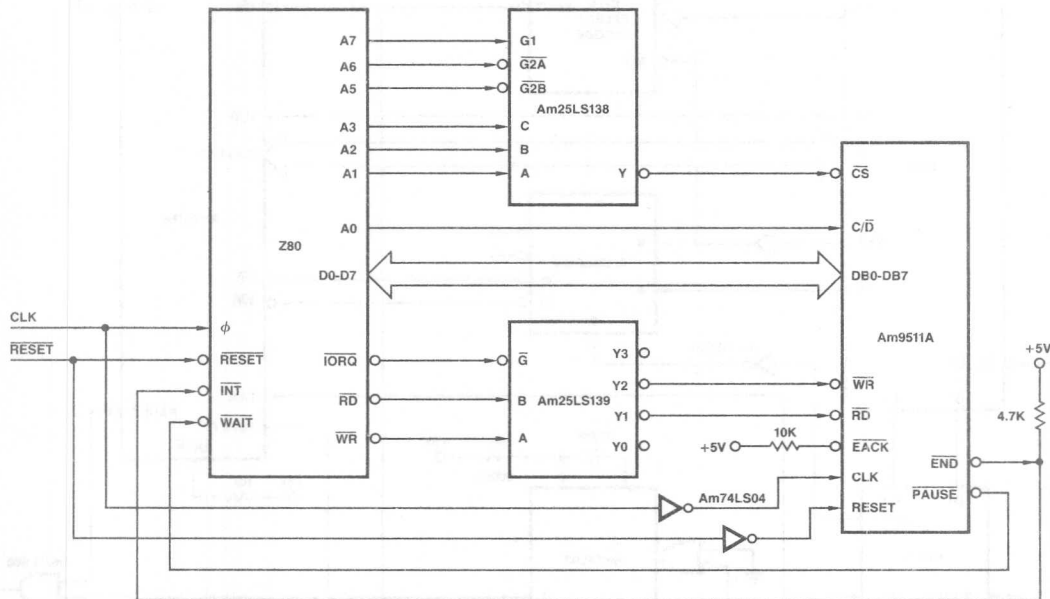
In a typical Am8085A system, the system clock rate is 3MHz and requires the use of an Am9511A-1 for this interface.

The Am8085A has an early ready set up window as compared to Am9080A systems. Therefore, the PAUSE signal cannot be directly connected to the READY input of the Am8085A. If it were, the READY line would be activated too late for the Am8085A to go into the wait state. The two Am74LS74 D flip-flops are used to

force one wait state whenever the Am9511A is accessed. After the first wait state, the Am74LS74's \overline{Q} output is reset and the PAUSE of the Am9511A-1 controls any additional wait states, if necessary. The example shows an interrupt driven interface. At the end of every Am9511A-1 operation, the \overline{END} signal goes low and activates the restart input to the Am8085A.

Am9511A to Z80 Interface

CIRCUIT DIAGRAM:



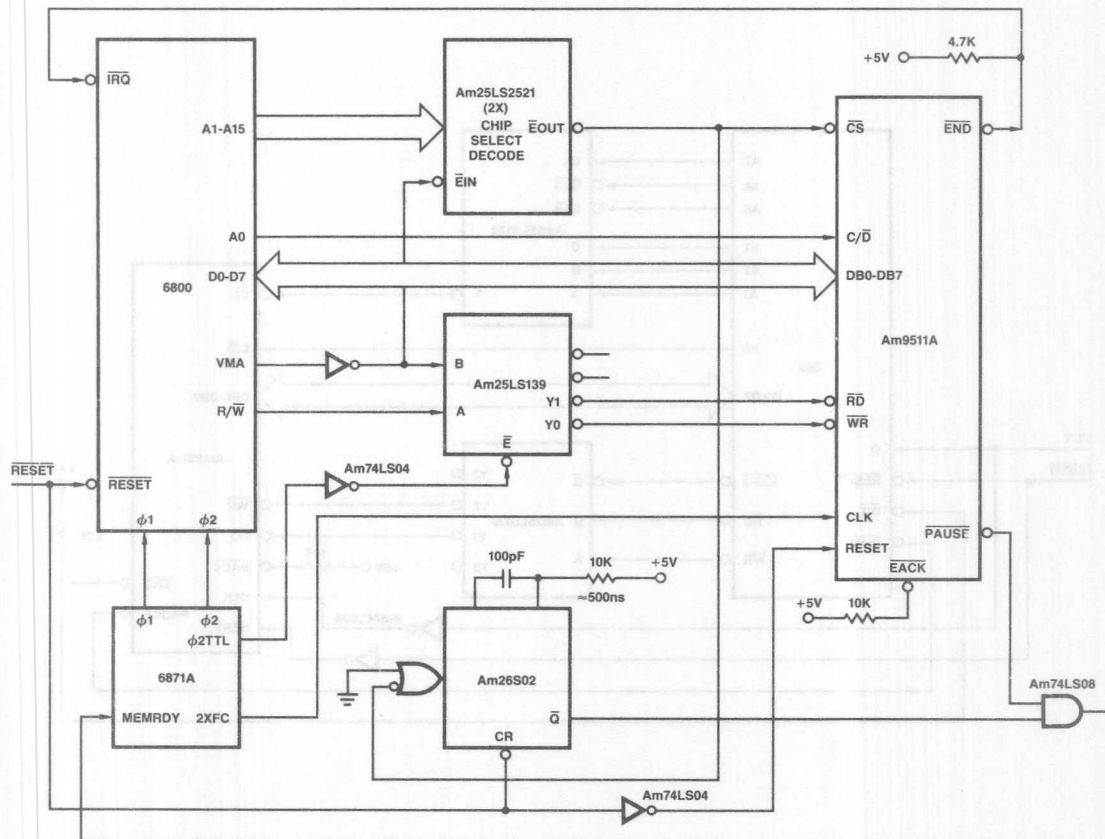
DESCRIPTION OF INTERFACE:

The Am9511A responds to a Data Read, Data Write or Command Write Request received while the Am9511A is busy by pulling the PAUSE output LOW. This causes the Z80 to enter a wait state until the Am9511A has finished the current transaction. It should be noted that the Am9511A outputs a LOW on PAUSE 150nsec

(max) after $\overline{\text{RD}}$ and $\overline{\text{WR}}$ has become active. The $\overline{\text{PAUSE}}$ remains LOW for (3.5 TCY + TON) nsec minimum during Status Read. (TCY is the clock period.) Therefore, the Z80 will insert one to two extra wait states. This interrupt-driven interface is straightforward and requires a minimum amount of external logic.

Am9511A to 6800 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

In this example, the Am9511A must be accessed through a memory map technique since the 6800 does not have explicit I/O instructions. The decoder is strobed by VMA to produce a glitch-free output.

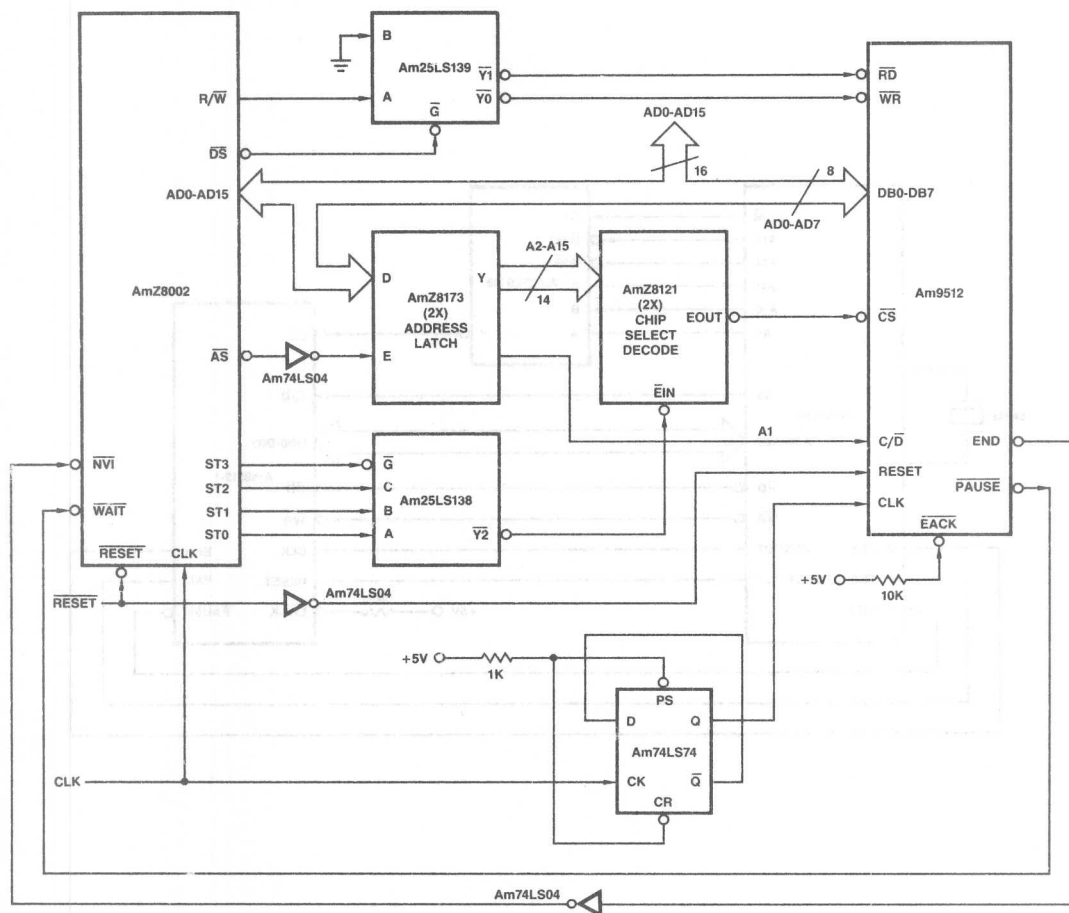
Since the Am9511A has a relatively long read access, the RD pulse must be stretched. This is done using the 6871A clock driver. The memory input stretches the ϕ_2 clock's HIGH time without affecting the 2XFC clock that is used as a clock for the Am9511A. When a \overline{CS} to the Am9511A is decoded, an Am26S02

one-shot is triggered and pulls the memory READY line LOW for approximately 500nsec. The one-shot is necessary because PAUSE will not go low soon enough to stretch ϕ_2 in the current cycle.

The programmer must take caution not to perform operations other than a status read while a current command is still in progress to avoid producing a PAUSE output longer than 4500nsec. The 6800 is a dynamic device and the clock input must not be stopped for more than 4500nsec.

Am9512 to AmZ8002 Interface

CIRCUIT DIAGRAM:

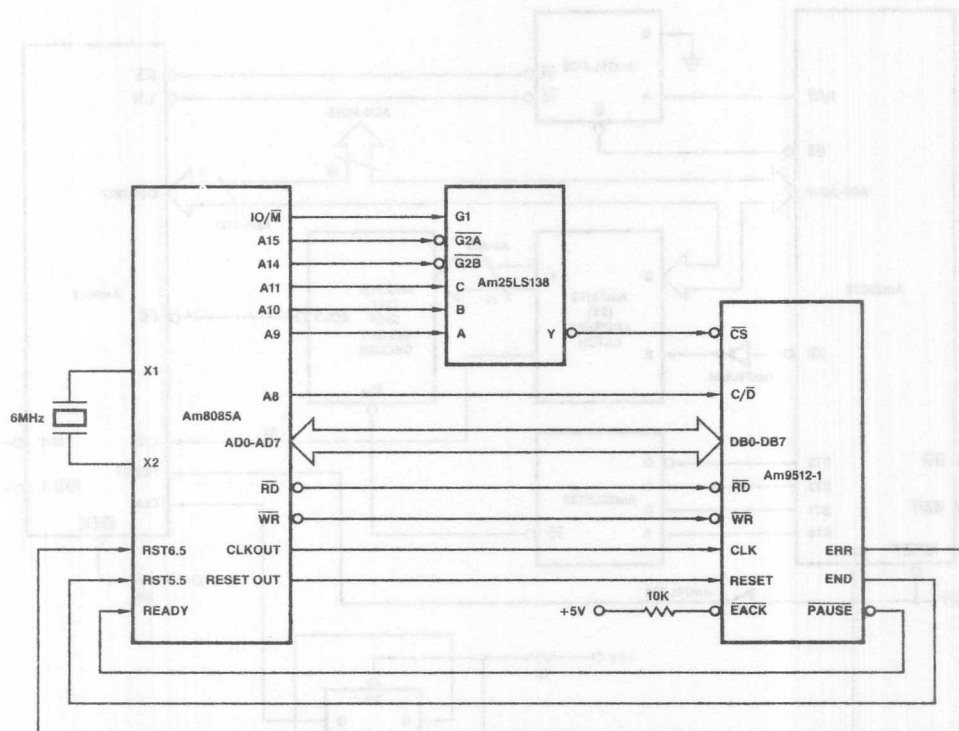


DESCRIPTION OF INTERFACE:

This interface is similar to the Am9511A to AmZ8002 interface except that the PAUSE output can be connected directly to the WAIT input of the CPU. As in the Am9511A interface, the 4MHz CPU clock is divided by the Am74LS74 flip-flop to present a 2MHz clock to the Am9512.

Am9512-1 to Am8085A Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

This example shows that the interface to the Am8085A is straightforward, requiring no additional logic. Since the typical Am8085a system runs at 3MHz, the Am9512-1 is used.

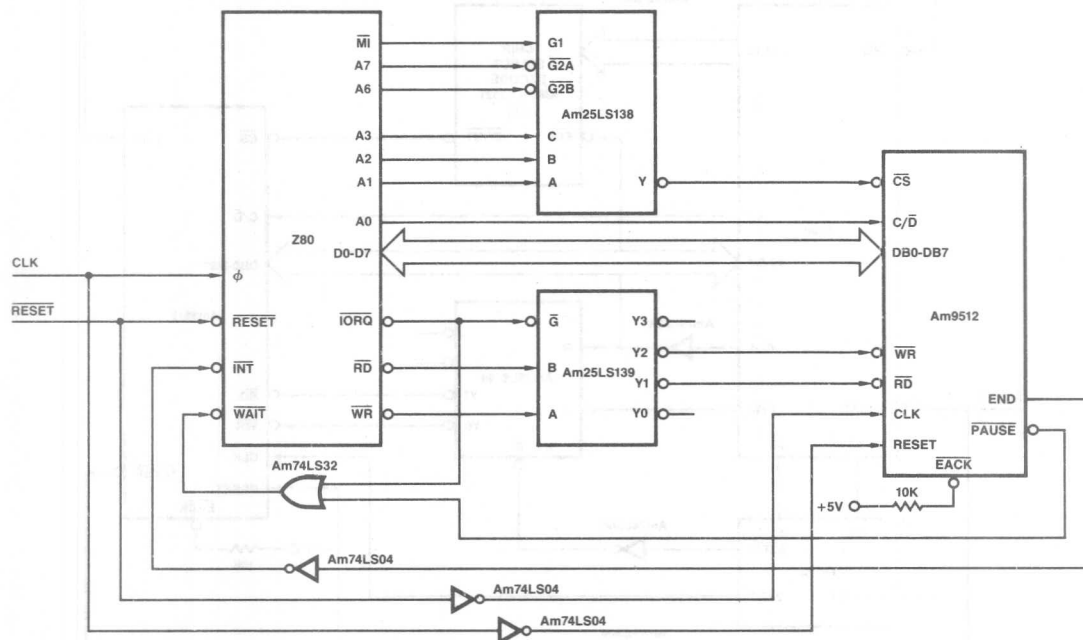
The ERR/END outputs are connected to separate interrupt inputs

so that the CPU can identify the source of the interrupt without reading the status register of the Am9512-1.

The chip select decoder is strobed with the IO/M signal thus enabling the Am9512-1 only during I/O operations.

Am9512 to Z80 Interface

CIRCUIT DIAGRAM:



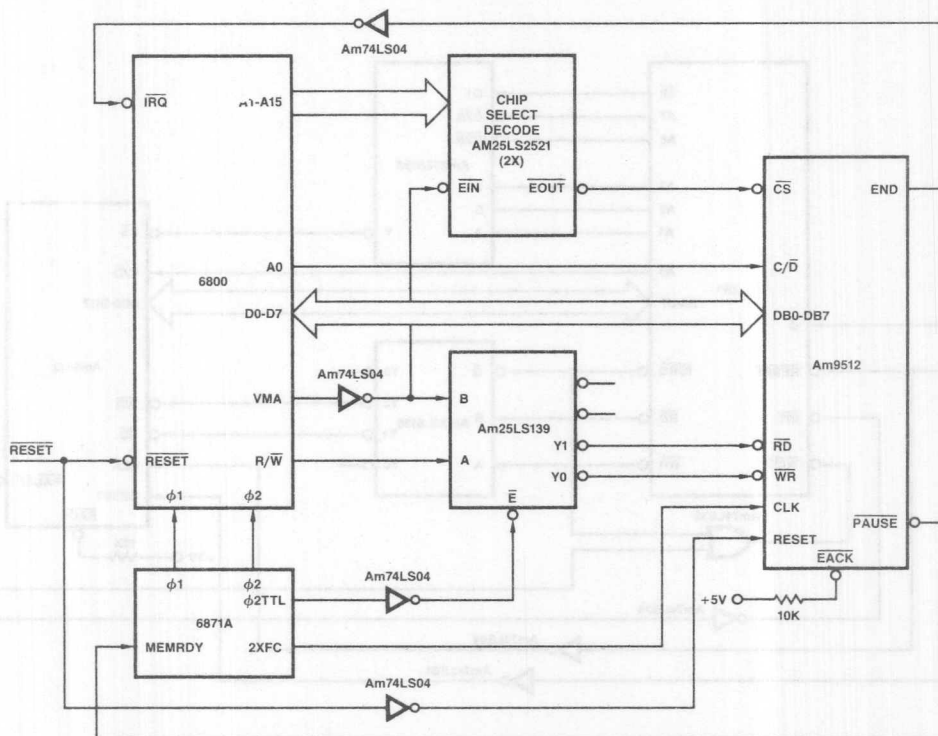
DESCRIPTION OF INTERFACE:

In this interface, two more additional gates are needed than the number required for the Am9511A interface. An inverter is added to the Interrupt Request (INT) line because the polarity of the END signal is different. The Am74LS32 is added in the WAIT line to insure the Am9512 PAUSE will go LOW whenever the Chip Select on the Am9512 goes LOW. In the circuit above, the Chip Select input can go LOW during the second or third cycle of an

instruction when the memory address matches the Am9512 I/O address. If the OR Gate is omitted, the WAIT input on the Z80 will go LOW and the system will be forced into a deadlocked state. The Chip-Select Decoder in this example is strobed with M1. This is done to guarantee a Chip Select on every I/O cycle and it also prevents the Chip Select from going LOW during an interrupt acknowledge cycle.

Am9512 to 6800 Interface

CIRCUIT DIAGRAM:



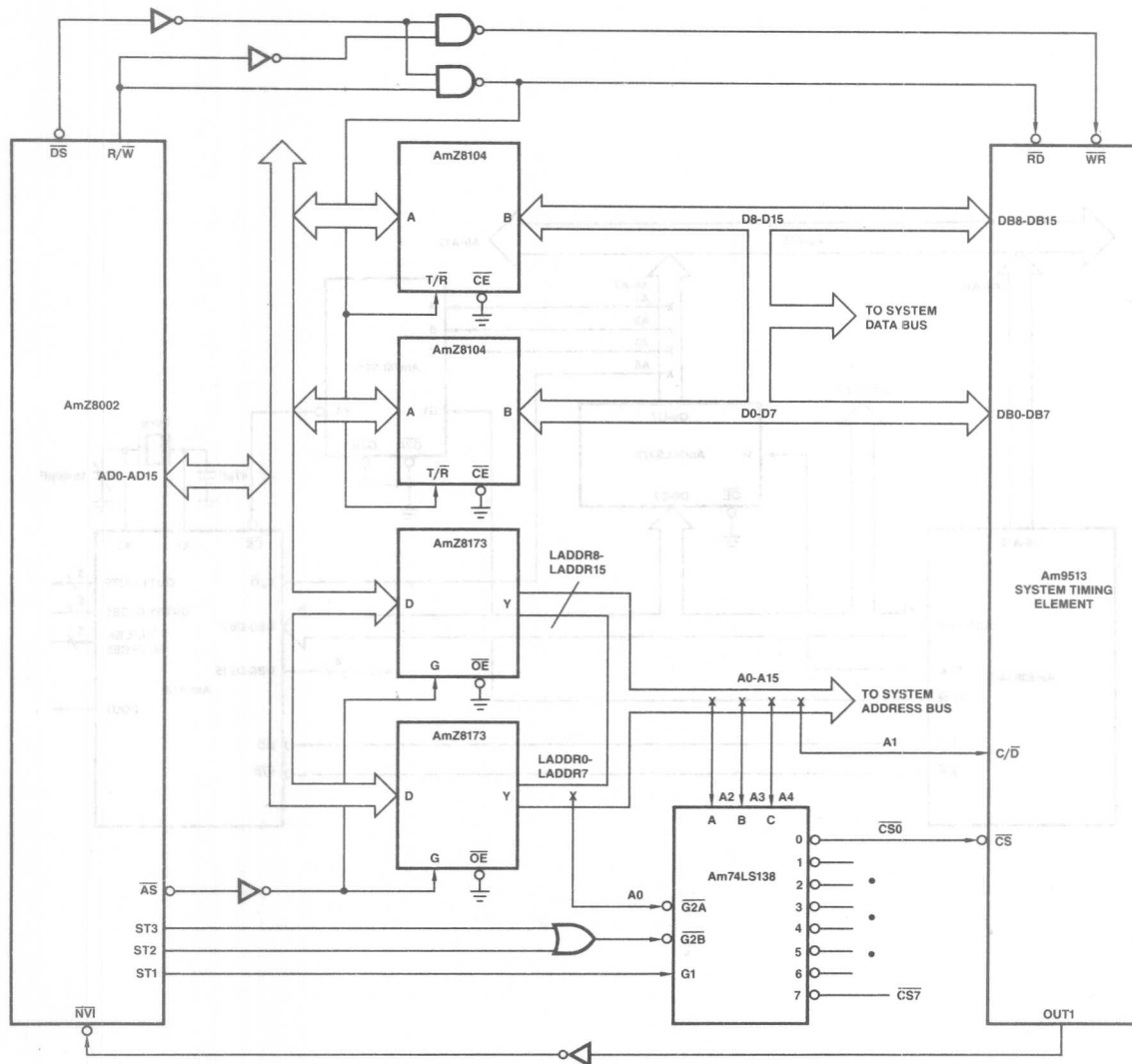
DESCRIPTION OF INTERFACE:

This interface is similar to the Am9511A interface with the exception of the one-shot. Since the $\overline{\text{PAUSE}}$ output from the Am9512 follows the $\overline{\text{CS}}$ instead of $\overline{\text{RD}}$ or $\overline{\text{WR}}$. The Memory Ready

(MEMRDY) signal can be directly driven by the $\overline{\text{PAUSE}}$ output. The only other addition is the inverter between the END output of the Am9512 and the $\overline{\text{IRQ}}$ input.

Am9513 to AmZ8002 Interface

CIRCUIT DIAGRAM:



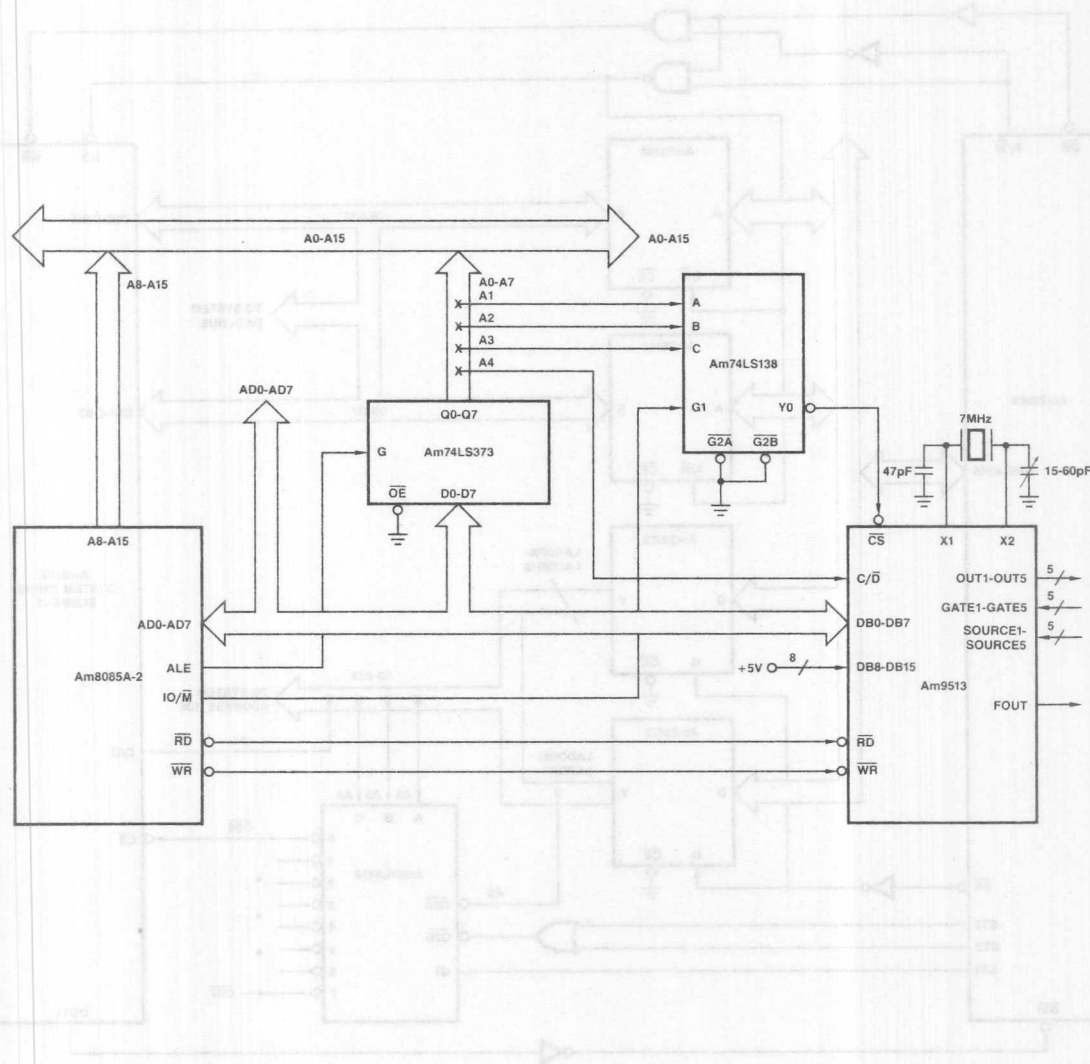
DESCRIPTION OF INTERFACE:

In this example, the Am9513 is connected to the 16-bit bus of the AmZ8002 via two AmZ8104 transceivers. Real-time interrupt capability is provided by connecting the Am9513 OUT1 output to the AmZ8002 NVI input. The address is latched into the AmZ8173, from the multiplexed bus during an Address Strobe

(AS). The inputs to the decoder come from the status output of the AmZ8002. The decoded status represents a normal I/O transaction. The RD and WR input signals of the Am9513 are generated by the R/W and D/S outputs on the AmZ8002.

Am9513 to Am8085A Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

In this example, the Am9513 device address can be easily located into either the memory or I/O space by selecting the desired logic enable level from the Am8085A IO/M line to the decoder.

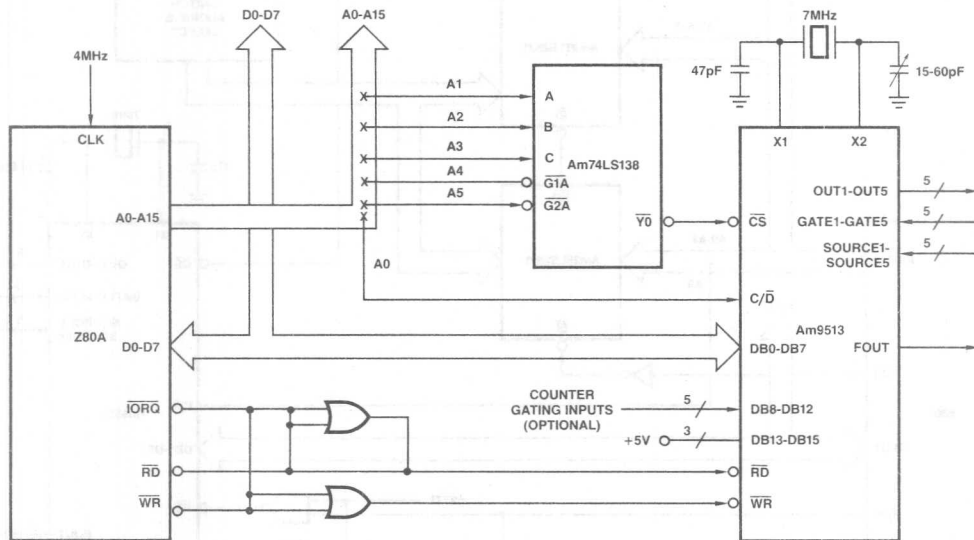
The Am25LS373 transparent 8-bit latch is not required to interface the Am9513 to the Am8085A. The latch is shown since

most systems require some form of demultiplexing for the low order address bits to decode the device address.

The DB8-15 data bus lines of the Am9513 should be tied HIGH, unless some of the pins are used as auxiliary gating inputs to the Am9513.

Am9513 to Z80A Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

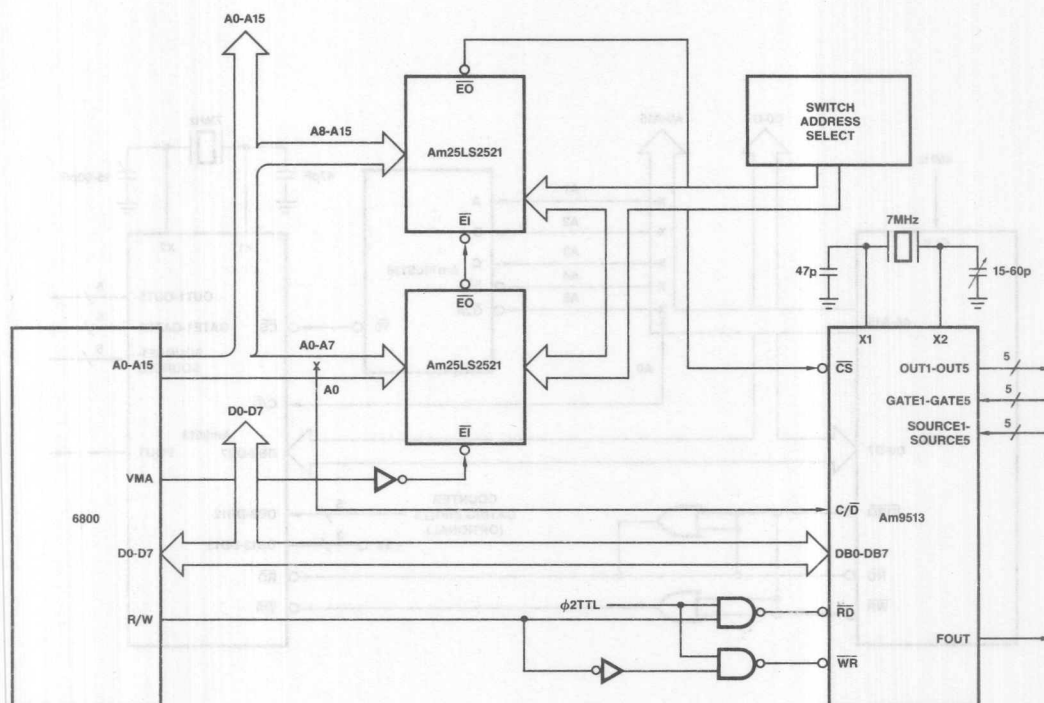
Gating \overline{RD} and \overline{WR} with \overline{IORQ} from the Z80A allows mapping the Am9513 into the I/O space.

The least-significant address line (A_0) is used to select the command or data port on the Am9513. The DB0-7 data bus lines of the

Am9513 interface directly to the Z80A data bus, while the DB13-15 lines should be tied HIGH. Pins DB8 through DB12 could be used as auxiliary gating inputs to the counters.

Am9513 to 6800 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

This interface is an example of how to use two 8-bit comparators for chip selection. The peripheral address can be modified via dip switches.

The least significant address bit (A_0) is used to select the Command/Data (C/D) port of the Am9513.

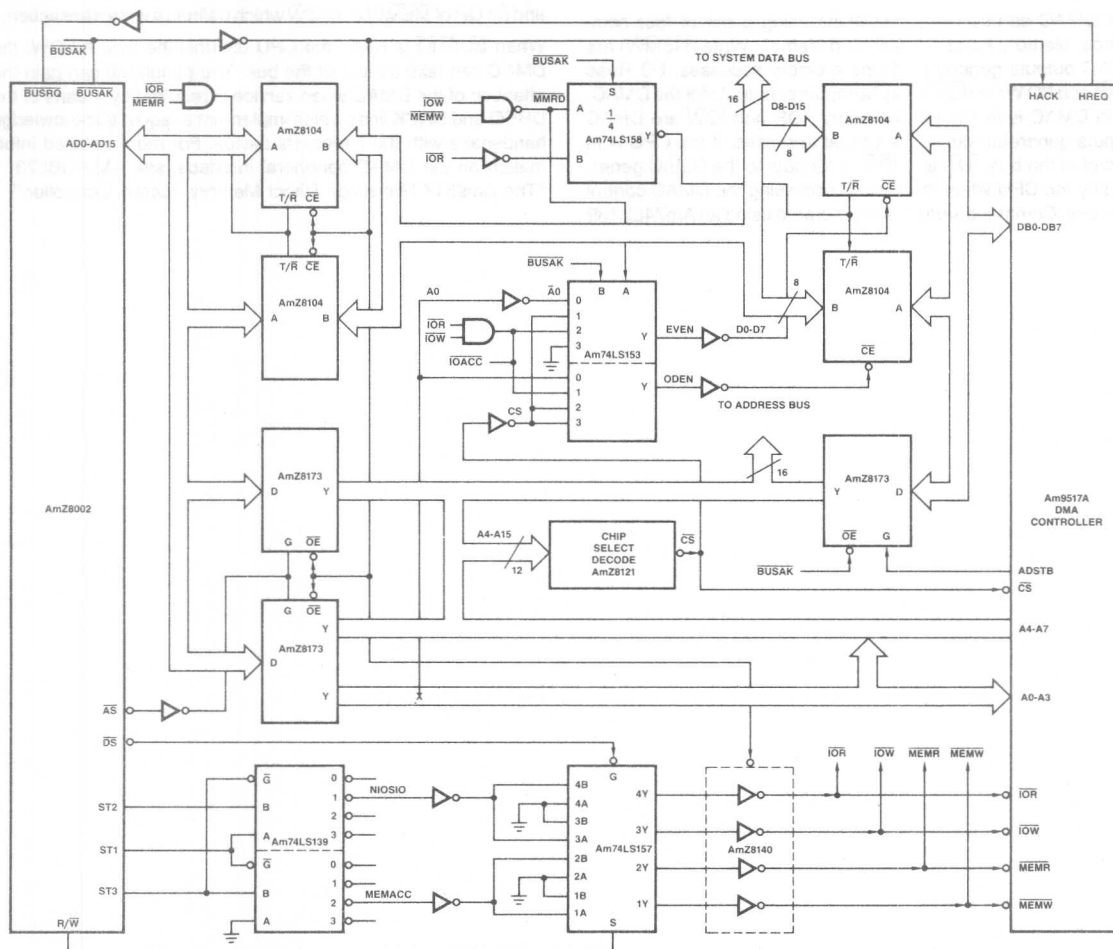
The VMA signal is used to assure non-spurious Chip Selects (i.e., during interrupts, DMA, etc). The DB0-7 data bus lines on the

Am9513 are used to interface the part to the 6800 data bus. The DB8-15 lines on the Am9513 must be tied HIGH to insure proper operation.

The \overline{RD} and \overline{WR} signals are decoded from the 6800 R/W line and are gated with the ϕ_2 TTL clock high signal in order to assure valid data transfers.

Am9517A to AmZ8002 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

The Am9517A DMA Controller can provide some useful advantages to an AmZ8002 system, both in terms of increased throughput and reduced latency in responding to peripheral requests for attention. The DMAC performs transactions between memory and I/O (or memory-to-memory) by gaining bus master-ship from the CPU. The DMAC then controls the bus by generating the necessary control and timing signals.

The bus exchanges between the CPU and DMAC are controlled by the Hold Request (HREQ) and Hold Acknowledge (HACK)

signals at the DMAC and the Bus Request ($\overline{\text{BUSRQ}}$) and Bus Acknowledge ($\overline{\text{BUSAK}}$) lines from the CPU.

The DMAC has separate 8-bit data and 8-bit address busses. The 8-bit address bus is used for outputting the lower half of the 16-bit memory address. The 8-bit data bus is buffered by two AmZ8104 transceivers, which fan the byte data to both halves of the 16-bit system bus during a DMAC output transaction. Similarly the buffers are used to steer either the upper or lower half of the bus onto the eight data lines of the DMAC during a DMAC input

transaction. With this configuration, memory-to-memory transfers, which are inherently implemented using the DMAC dataflow, can take place between any two memory byte locations.

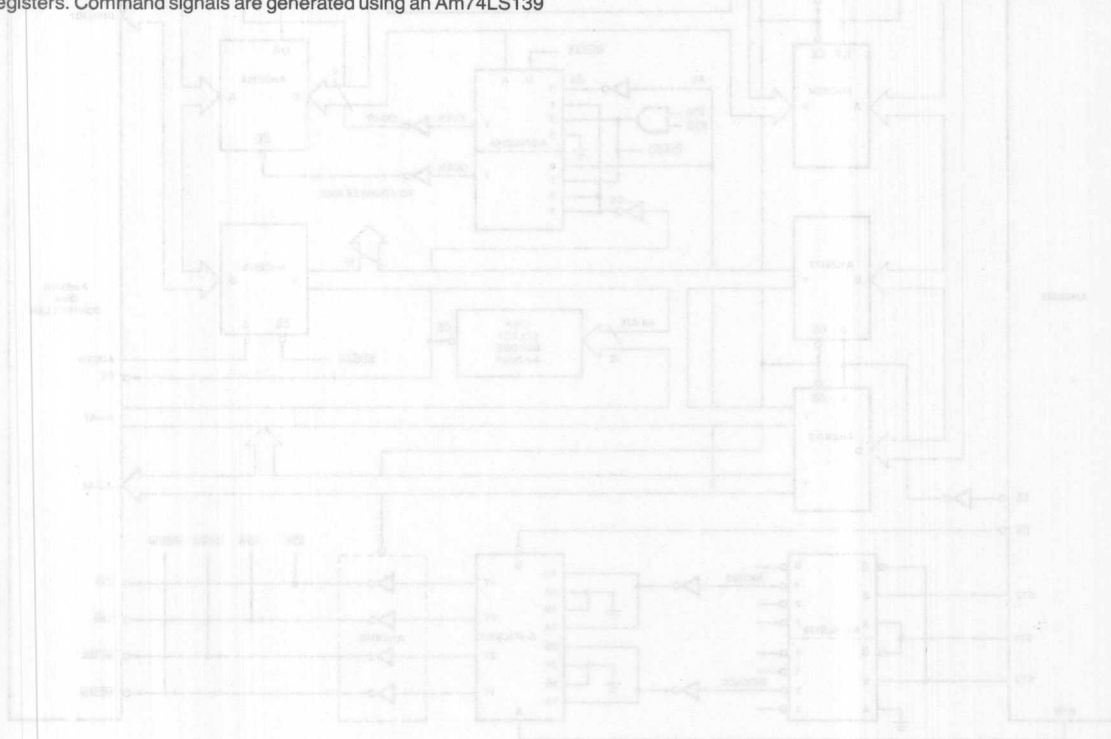
The Chip Select is generated from a comparison of the most-significant 12 address bits, since the least significant four address bits are input to the DMAC. The CS is only enabled when an I/O transaction is underway.

The DMAC carries out transactions using a set of four commands. Memory Read (MEMR) and Memory Write (MEMW) are DMAC outputs generated during memory accesses. I/O Read (IOR) and I/O Write (IOW) may be inputs or outputs for the DMAC. If the DMAC is in control of the bus, IOR and IOW are DMAC outputs generated during peripheral accesses. If the CPU is in control of the bus, IOR and IOW are inputs to the DMAC generated by the CPU when the latter is accessing the DMAC control registers. Command signals are generated using an Am74LS139

dual 2-to-4 decoder cascaded with an Am74LS157 quad 2-to-1 multiplexer.

The control requirements for the DMAC address and data buffers are different for both cases and thus the final stage of control generation is implemented through an Am74LS153 multiplexer. The multiplexer select lines are driven from the CPU BUSAK line (which defines whether the CPU or DMAC is in control of the bus) and an OR of MEMW and IOW which defines a write transaction.

When BUSRQ is HIGH the CPU controls the bus. If LOW, the DMAC can take control of the bus. The peripheral can gain the attention of the DMAC when service is required by means of the DREQ and DACK lines which implement a request/acknowledge handshake with the peripheral device. For more detailed information on the DMAC peripheral interface, see AM-PUB073 - "The Am9517 Multimode Direct Memory Access Controller."



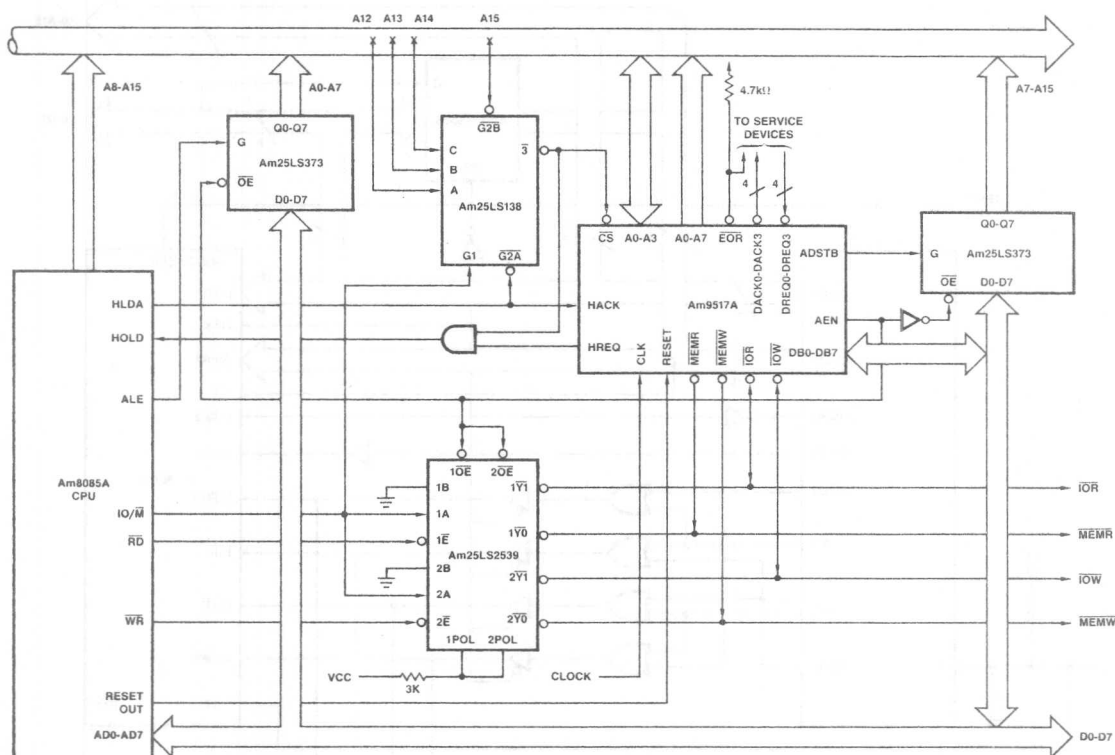
DESCRIPTION OF INTERFACE

The Am9517A DMAC Controller provides some useful external signals to the CPU and the Bus Request (BUSRQ) and Bus Acknowledge (BUSAK) lines from the CPU. The DMAC uses a 16-bit address bus and a 16-bit data bus. The 16-bit address bus is used for outputting the lower half of the 16-bit memory address. The 16-bit data bus is buffered by two 8-bit buffers, which can be byte data or half-word or the 16-bit data bus during a DMAC output transaction. Similarly, the system bus used to transfer data to or from the bus during a CPU data transfer of the DMAC during a CPU data transfer.

The Am9517A DMAC Controller provides some useful external signals to the CPU and the Bus Request (BUSRQ) and Bus Acknowledge (BUSAK) lines from the CPU. The DMAC uses a 16-bit address bus and a 16-bit data bus. The 16-bit address bus is used for outputting the lower half of the 16-bit memory address. The 16-bit data bus is buffered by two 8-bit buffers, which can be byte data or half-word or the 16-bit data bus during a DMAC output transaction. Similarly, the system bus used to transfer data to or from the bus during a CPU data transfer of the DMAC during a CPU data transfer.

Am9517A to Am8085A Interface

CIRCUIT DIAGRAM:

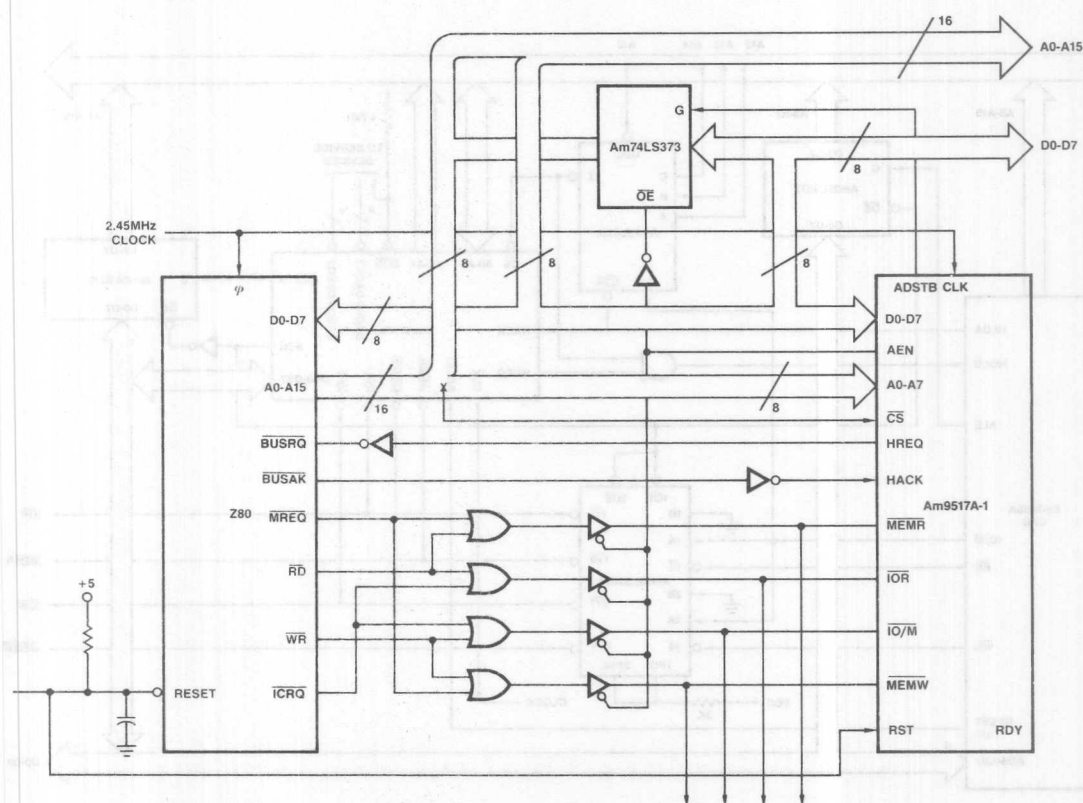


DESCRIPTION OF INTERFACE:

The Am9517A DMA controller is conveniently configured in an Am8085A microprocessor system. The multimode DMA controller issues a Hold Request whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A takes control of the address bus, data bus and control bus. The Am25LS2539 dual one-of-four decoder with three-state outputs easily decodes the IO and MEM space control lines from the processor, and also isolates the Am8085A processor when DMA

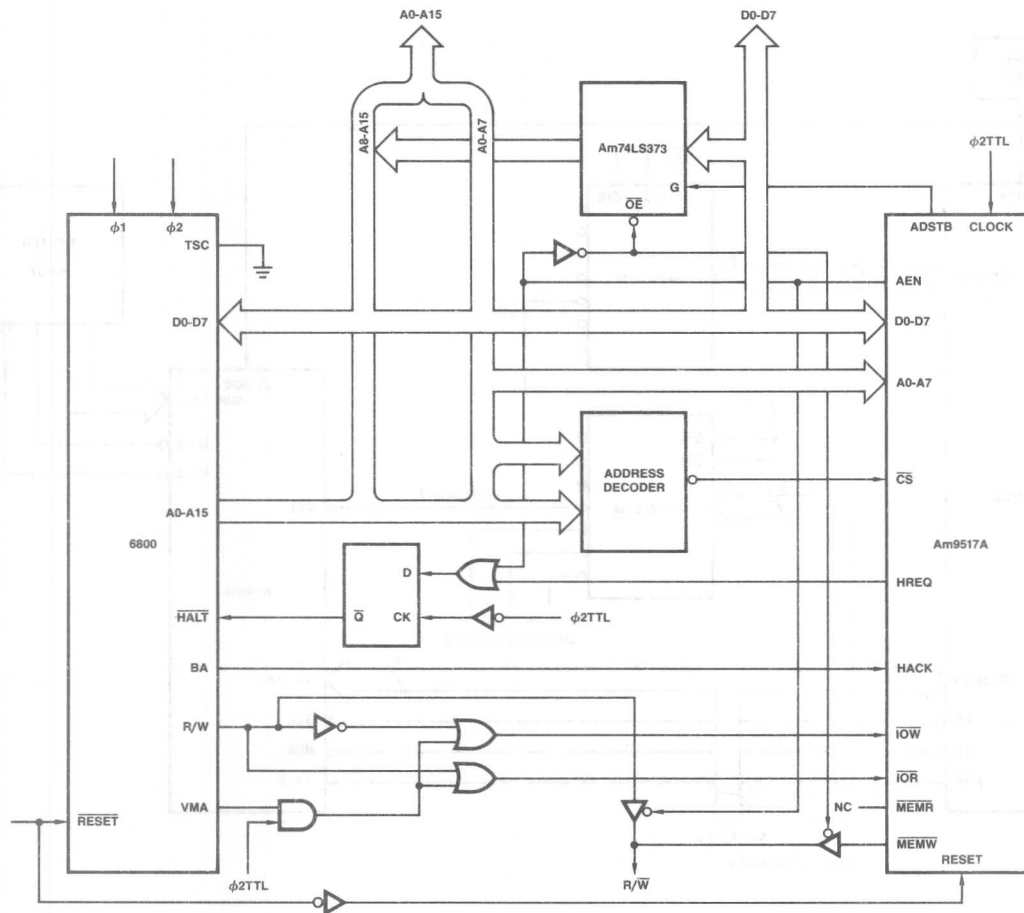
operations are in process. The address for the first transfer operation comes out of the DMA Controller in two bytes, the least-significant eight bits on the eight address outputs and the most-significant eight bits on the data bus. The contents of the data bus are conveniently latched into the Am25LS373 register to complete the full 16 bits of the address bus. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least-significant address byte.

Am9517A-1 to Z80 Interface



Am9517A to 6800 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

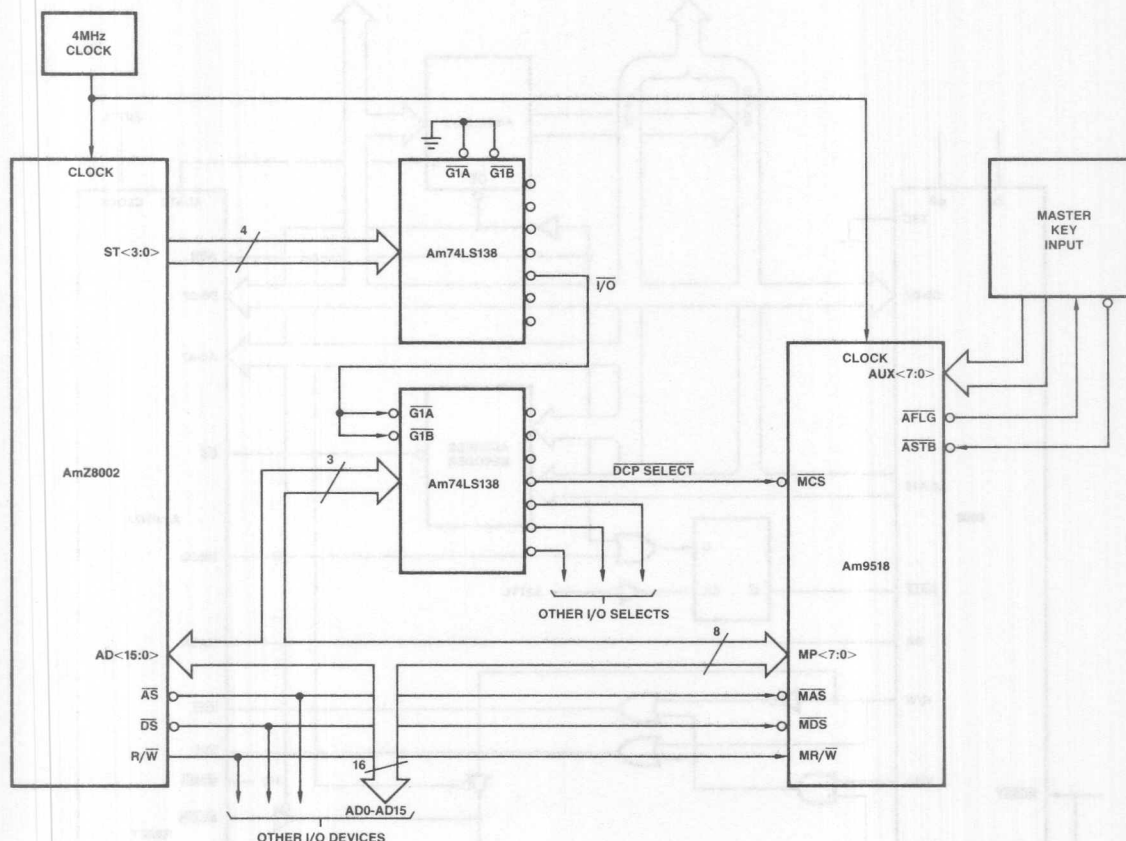
The Am9517A DMA controller is conveniently interfaced to a 6800 microprocessor system. An Am74LS373 type 8-bit latch holds the high order A8 to A15 addresses during DMA transfers.

A D-type flip-flop synchronizes the Am9517A Hold Request (HREQ) signal with the ϕ_2 clock "low" as required by the 6800. The HREQ and AEN (Address Enable) outputs from the Am9517A are ORed together to simplify giving back the bus to the

6800 at the end of the DMA operation because the DMA is still active on the bus for about one clock period following HREQ going inactive. This method also allows unlimited periods of DMA transfers, since the 6800 is not stopped by using its TSC input. Usage of the TSC input requires holding the ϕ_1 and ϕ_2 clocks to the 6800 high and low, respectively, for not longer than 4500nsec (1MHz 6800).

Am9518 to AmZ8002 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

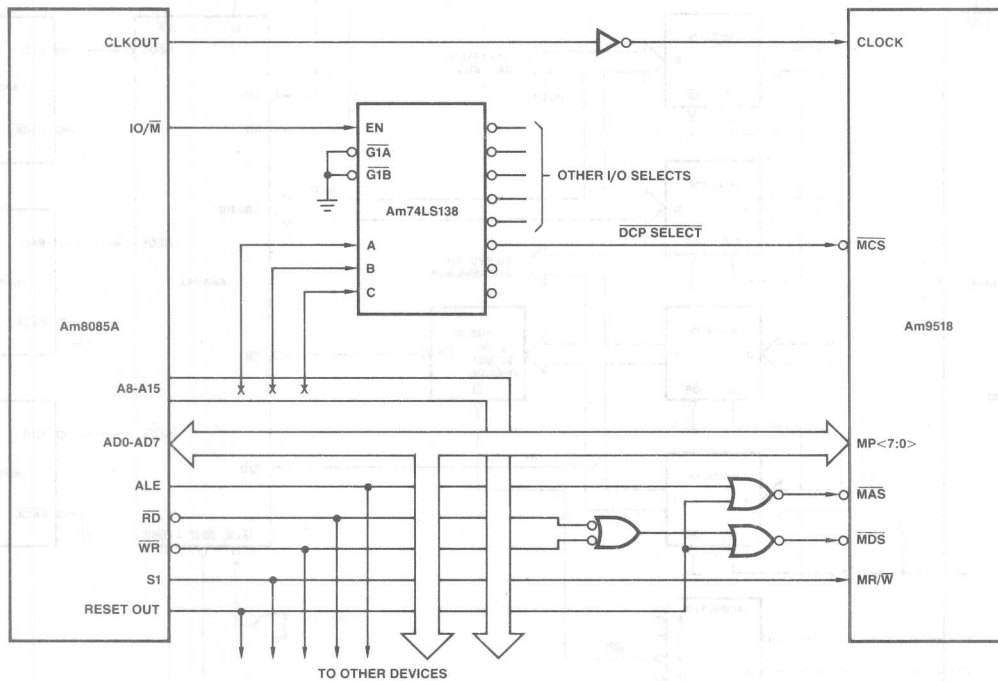
The Am9518 Data Cyphering Processor interfaces very conveniently to an AmZ8002 bus structure. The I/O address space access is decoded from the ST0 to ST3 status pins of the AmZ8002 processor while, simultaneously, the three upper address lines (A₁₃-A₁₅) are being decoded by another Am74LS138

type decoder for activation of the Am9518 chip select input. The example is sufficient for a small system configuration.

The AD0-15 bus lines are shown as an 8-bit interface to the Am9518.

Am9518 to Am8085A Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

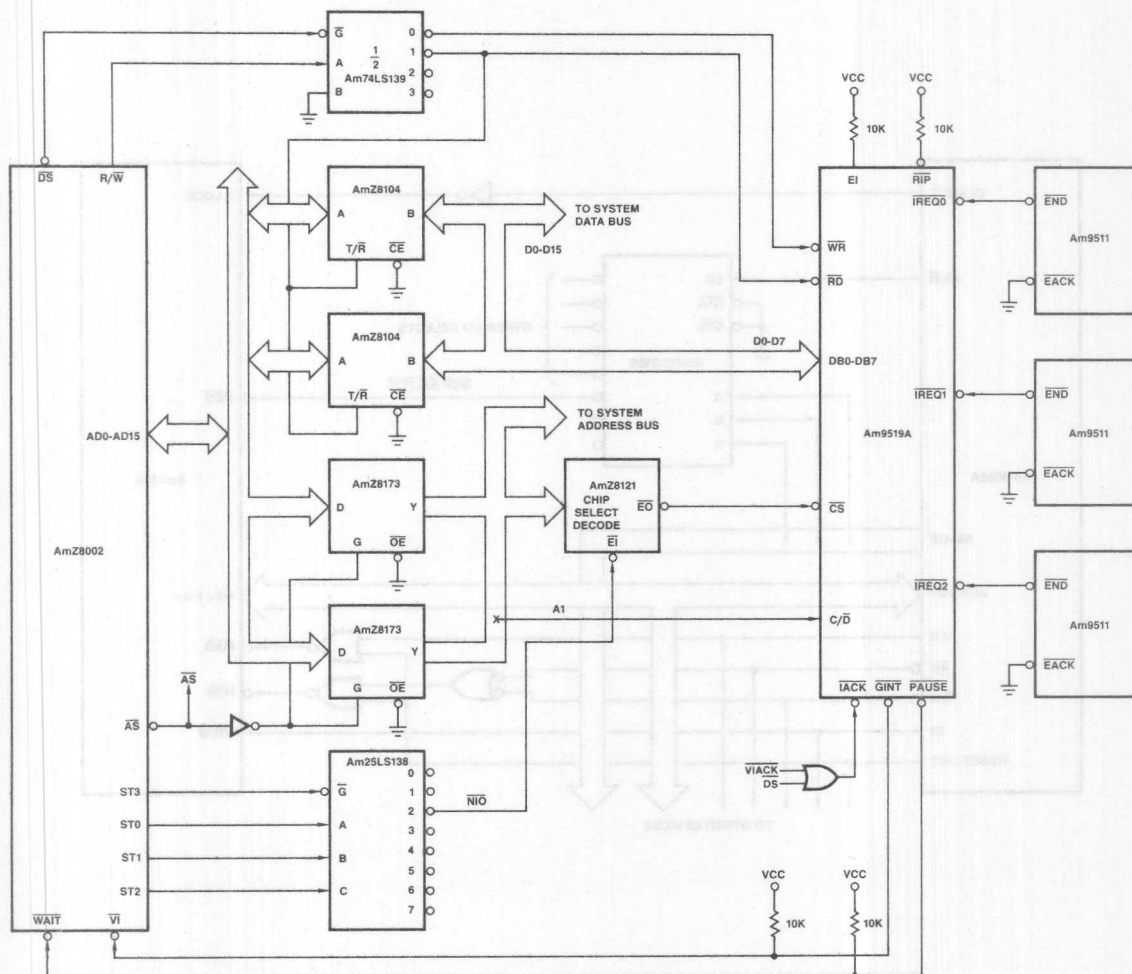
The Am9518 Data Cyphering Processor interfaces very simply to the Am8085A microprocessor. The Am9518 is shown mapped in an IO address space with its Chip Select input being decoded from three suitable lines of the A_8 to A_{15} address and AD_0 to AD_7 address/data bus. The ALE output generates the required AS strobe, while either the \overline{RD} or \overline{WR} outputs generate the DS

strobe. The S1 output of the Am8085A is used to specify a read or write operation.

The Am8085A microprocessor cycle time should not be greater than 2MHz in this configuration in order to allow Am9518 to synchronize to the processor.

Am9519A to AmZ8002 Interface

CIRCUIT DIAGRAM:



DESCRIPTION OF INTERFACE:

The application involving the connection of the Am9519A to the AmZ8002 (pg. 2-1) highlights the limitations of a direct interrupt connection to the CPU. The inclusion of the Am9519A Interrupt Controller in the system relieves the CPU of a large part of this task and also serves to prioritize multiple interrupts.

The AmZ8173 latches and AmZ8104 bidirectional buffers generate separate address and data busses from the CPU address/data bus. The latched address is input to two 8-bit

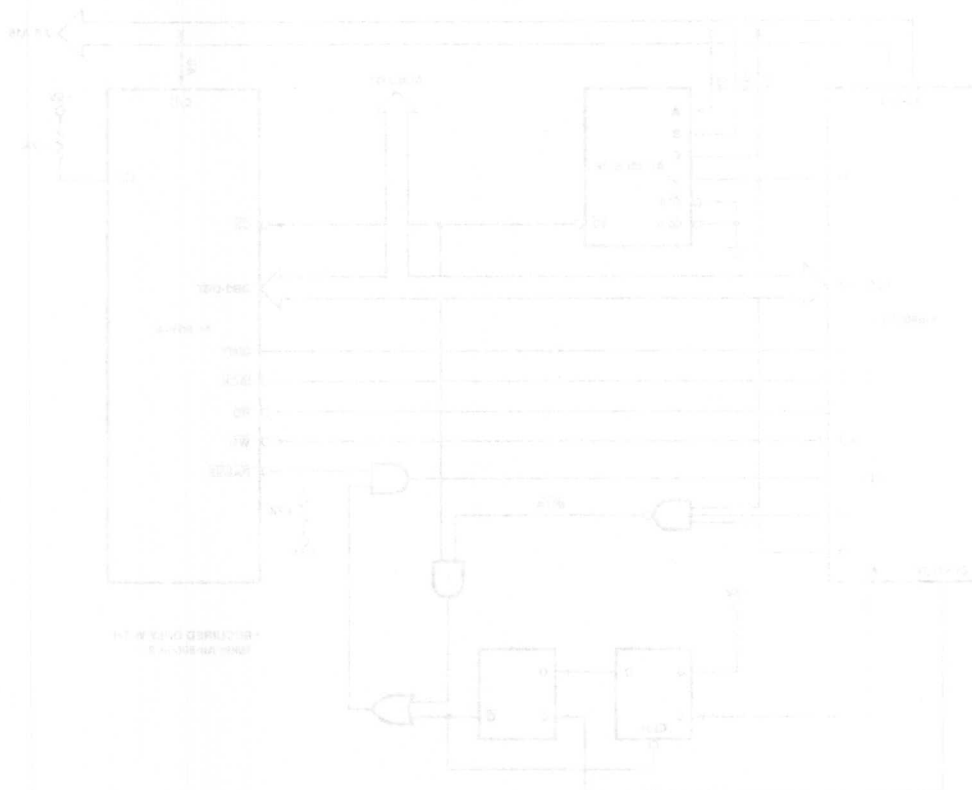
comparators to generate a Chip Select (\overline{CS}) for the Am9519A if a normal I/O transaction is being executed. Latched address bit A1 drives the C/D input to the Am9519A.

The lower half of the buffered data bus is shown driving the bidirectional data bus of the Am9519A. This is applicable with the choice of odd port addresses, should byte I/O operations be executed. The read and write commands required by the Am9519A (\overline{RD} and \overline{WR} respectively) are generated with half of

an Am74LS139 decoder. \overline{DS} is applied to the enable input and $\overline{R/W}$ to the least-significant select line. This is an alternative solution to a discrete gate implementation.

The Am9519A makes a vectored interrupt request to the CPU using the Group Interrupt line (\overline{GINT}) which should be a LOW active output from the Am9519A. \overline{GINT} is reset by the CPU Vectored Interrupt Acknowledge (\overline{VIACK}). The latter is decoded from the status lines and is or'ed with \overline{DS} to generate \overline{IACK} to the Am9519A.

The Am9519A should be programmed to respond to a single interrupt acknowledge, which in turn results in the transfer of one byte of interrupt status to the CPU. This is sufficient since the vectored interrupt mechanism in the CPU requires only one byte of status to form the vector. The interrupts input to the Am9519A from the devices may be levels or pulses. If levels are employed, then some form of request reset will have to be included. In the implementation shown, pulsed interrupt requests are assumed.

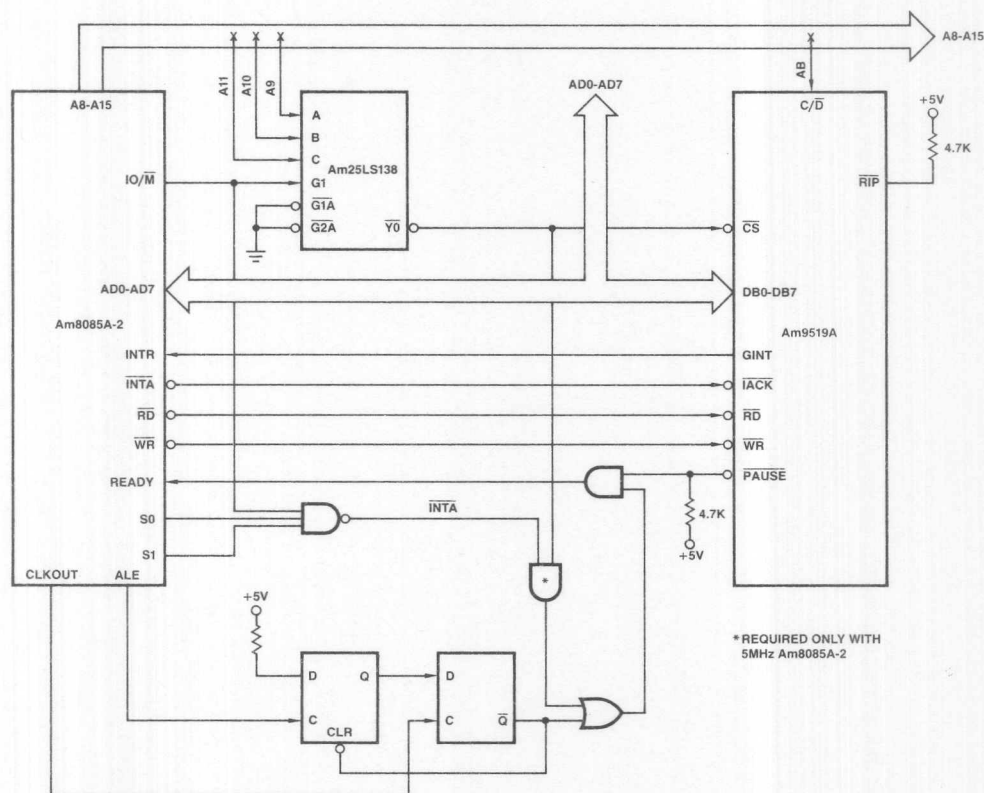


100ns maximum delay. READY, Am9519A PAUSE output taken over control line ready the following 100ns. Am9519A will be released only when the Am9519A-2 is ready. The wait state is needed because the minimum RD and WR pulse width of the Am9519A-2 is 200ns while the Am9519A-1 is 100ns. The Am9519A-2 is 200ns minimum and the Am9519A-1 is 100ns minimum.

DESCRIPTION OF THE INTERFACE
The two 16-bit bus lines to the processor with 20, 21 and 22 are used to transfer the data between the processor and the Am9519A. The processor is required to send one byte of data to the Am9519A (data) and one byte of data from the Am9519A to the processor (data). The Am9519A sends a READY signal to the processor on the rising edge of the clock. The processor is required to send a 100ns minimum pulse to the Am9519A on the rising edge of the clock. The Am9519A sends a 100ns minimum pulse to the processor on the rising edge of the clock. The Am9519A sends a 100ns minimum pulse to the processor on the rising edge of the clock. The Am9519A sends a 100ns minimum pulse to the processor on the rising edge of the clock.

Am9519A to Am8085 Interface

CIRCUIT DIAGRAM:



*REQUIRED ONLY WITH
5MHz Am8085A-2

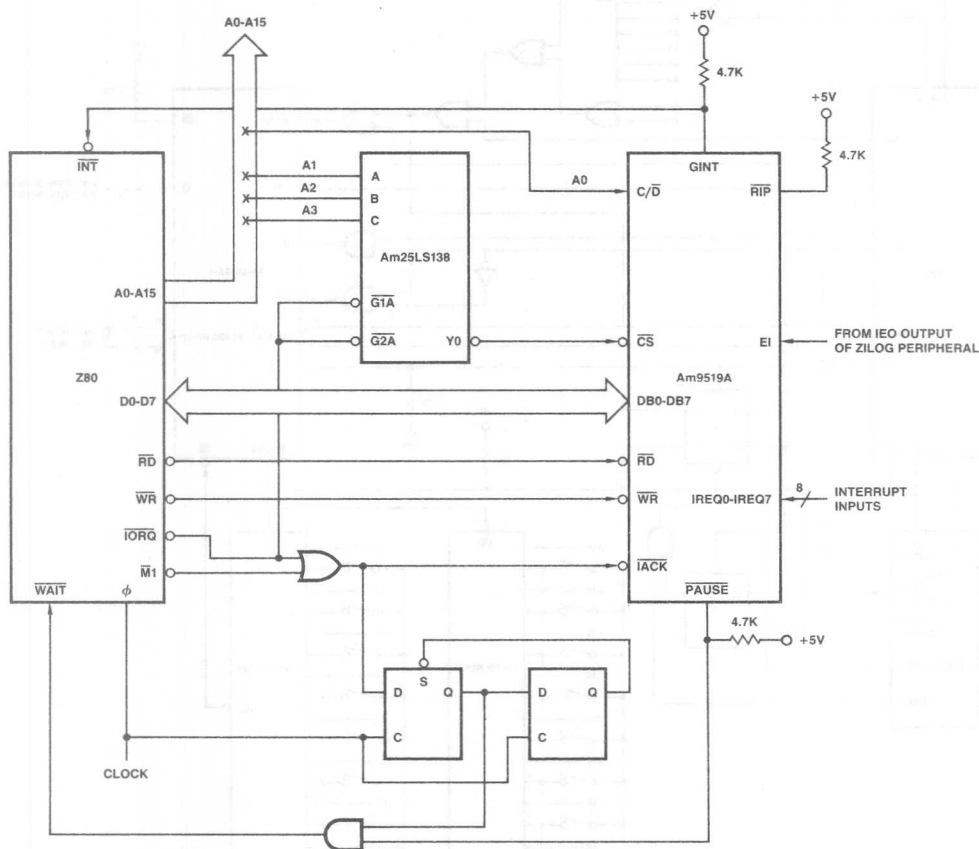
DESCRIPTION OF INTERFACE:

The two D-type flip-flops in conjunction with S0, S1 and IO/M insert a wait state to the Am8085A whenever the microprocessor acknowledges an interrupt. The circuitry is required since the Am8085A samples a READY (wait) signal on the rising edge of CLK OUT during T2, and ready requires a 100nsec minimum set up time prior to the rising edge of T2. The 3-input NAND gate predecodes an INTA active signal using the two status outputs (S0, S1) and the IO/M address pin. This is required since the INTA output from the Am8085A becomes valid too late to meet the

100nsec setup time of READY. The Am9519A's PAUSE output takes over control of the ready line following T2.

Insertion of a wait during the read and write operation to the Am9519A with CS is required only when the 5MHz Am8085A-2 part is used. The wait state is needed because the minimum RD and WR pulse width of the Am8085A-2 is 230nsec, while the Am9519A-1, as example, requires 250nsec minimum and the Am9519A, 300nsec minimum.

Am9519A to Z80 Interface



DESCRIPTION OF INTERFACE:

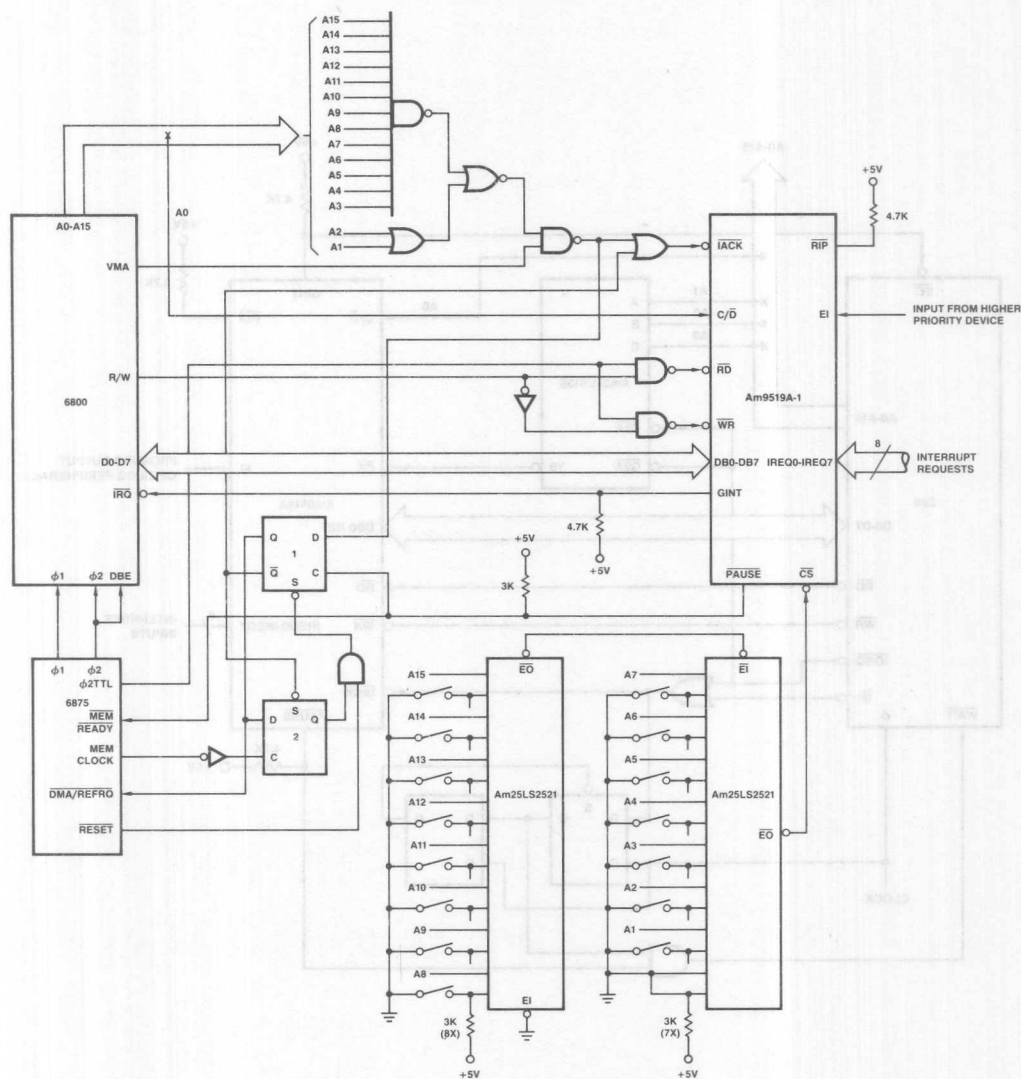
The Am9519A can be configured as the sole interrupting device to the Z80 or in conjunction with other devices (such as the SIO, CIO, CTC, etc.) in a daisy chain interrupt nest. When the Am9519A is part of a nested daisy chain, it must be the lowest priority device because the Am9519 activates its Enable Out (EO) output after receipt of the interrupt acknowledge signal ($\overline{\text{IORQ}} \cdot \overline{\text{M1}}$) from the microprocessor, while the Zilog parts require their IEI (Interrupt Enable In) signal to setup at least 200nsec (2.5MHz Z80) prior to the Interrupt Acknowledge going active.

A wait state is inserted in the CPU whenever an Interrupt Acknowledge sequence begins since the Am9519A begins re-

solving its interrupt priorities after the interrupt acknowledge signal goes active. The flip-flops assert a wait state at the second TW falling clock edge and allow the $\overline{\text{PAUSE}}$ output to extend the wait state as required.

For the 4MHz Z80A CPU, a wait state must be asserted whenever the CPU performs a read or write operation to the Am9519A in order to meet the \overline{RD} and \overline{WR} pulse width requirements. Either an Am9519A or Am9519A-1 can be used with the Z80A.

Am9519A to 6800 Interface



DESCRIPTION OF INTERFACE:

The Am9519A interfaces to a 1MHz 6800 microprocessor by using two sets of address decoders and a scheme to insert wait states during both ϕ_1 and ϕ_2 of the clock. Comparators provide the means of memory mapping the Am9519A device in order to access its internal registers. The Am9519A requires only two addresses for access, therefore the least-significant bit, A0, of the address bus drives the Command/Data (C/D) input directly.

For interrupt acknowledges, the addresses FFF8 and FFF9 are decoded because the 6800 fetches its interrupt vector (PC ad-

dress or jump address) from the above memory locations. The output of two bytes from the Am9519 provides the proper jump address for each interrupting device.

The Am9519A begins resolving its highest priority interrupt following the active edge of Interrupt Acknowledge IACK which is the address FFF8 in a 6800 system, and requires a 500 to 1000nsec delay before the first vector address appears on the data bus. The delay is accomplished by stretching the ϕ_2 clock high via the PAUSE active signal to the MEM READY input of the

6875 clock device. The only constraint is for the $\overline{\text{PAUSE}}$ signal to be active before the rising edge of ϕ_2 clock. The Am9519A-1 device may be preferred because the $\overline{\text{PAUSE}}$ output goes active within 125nsec following $\overline{\text{IACK}}$ active vs. 175nsec for the Am9519.

Flip-flops are used to create a 500nsec minimum $\overline{\text{IACK}}$ HIGH pulse width between the rising edge of the first $\overline{\text{IACK}}$ active ($\overline{\text{PAUSE}}$ going inactive and ϕ_2 clock going LOW) and the falling

edge of the second $\overline{\text{IACK}}$ active. This is done by stretching the pulse width of ϕ_1 clock. The rising of $\overline{\text{PAUSE}}$ causes flip-flop 1 to assert a $\overline{\text{DMA/REF RQ}}$ to the 6875 before address FFF9 is valid on the 6800 address bus in order to meet the setup time requirement of the 6875. The falling edge of the next MEM clock (ϕ_2 TTL ungated) from the 6875 causes stretching of ϕ_1 clock HIGH to terminate, $\overline{\text{IACK}}$ to become active now, and a second valid byte to be present on the data bus during the ϕ_2 clock HIGH period.

SUMMARY

PROCESSOR/PERIPHERAL SELECTION CHART

Processor	Am8085A	Am8085A-2	AmZ8001/2	Z80A	6800
Clock Period	320ns	200ns	250ns	250ns	1.0 μ s
Clock Generator	N/A	N/A	AmZ8127	AmZ8127	N/A
Data Cypher. Processor	Am9518 (Note 3)	Am9518 (Note 3)	Am9518	(Note 4)	(Note 4)
System Timing Controller	Am9513	Am9513	Am9513	Am9513	Am9513
Arithmetic Processor	Am9511A-1	Am9511A-1	Am9511A-1	Am9511A-1	Am9511A
	Am9512-1 (Note 2)	Am9512-1 (Notes 1 and 2)	Am9512-1 (Note 1)	Am9512-1 (Note 1)	Am9512
Univ. Interrupt Controller	Am9519A-1	Am9519A-1 (Note 2)	Am9519A-1 (Note 2)	Am9519A-1 (Note 2)	Am9519A-1 Am9519A (Note 2)
Multimode DMA Controller	Am9517A-1	Am9517A-1	Am9517A-4	Am9517A-1	Am9517A
	Am9517A-4	Am9517A-4 (Notes 1 and 2)	(Note 2)	(Note 2)	

Notes: 1. Clock input cannot be driven directly from processor clock; peripheral required lower frequency (processor clock-2).

2. CPU/peripheral transfers require addition of wait state, using external circuitry.

3. Clock input cannot be greater than 2MHz in order to allow Am9518 synchronization.

4. Am9518 requires additional logic for demultiplexing.

Am9500 Family Data Sheets

Am9511A

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- Replaces Am9511
- Fixed point 16 and 32 bit operations
- Floating point 32 bit operations
- Binary data formats
- Add, Subtract, Multiply and Divide
- Trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation
- Float to fixed and fixed to float conversions
- Stack-oriented operand storage
- DMA or programmed I/O data transfers
- End signal simplifies concurrent processing
- Synchronous/Asynchronous operations
- General purpose 8-bit data bus interface
- Standard 24 pin package
- +12 volt and +5 volt power supplies
- Advanced N-channel silicon gate MOS technology
- 100% MIL-STD-883 reliability assurance testing

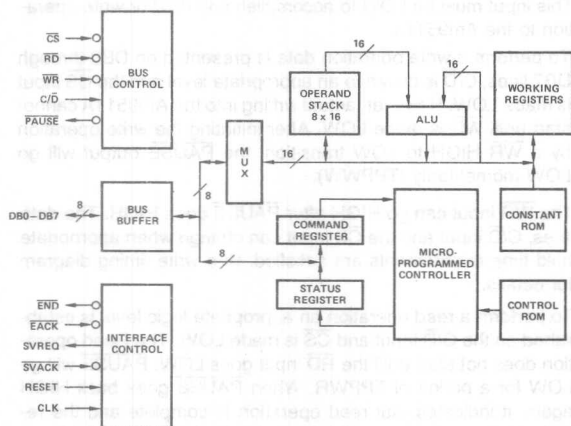
GENERAL DESCRIPTION

The Am9511A Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

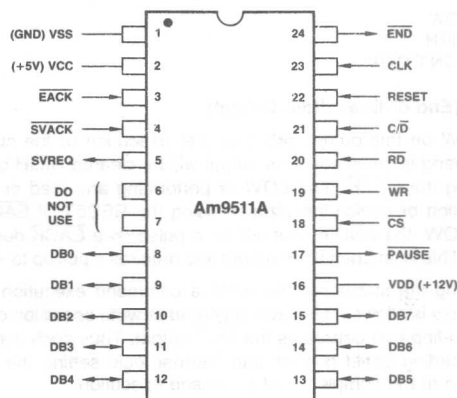
Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

BLOCK DIAGRAM



MOS-046

CONNECTION DIAGRAM Top View



Pin 1 is marked for orientation.

MOS-047

ORDERING INFORMATION

Package Type	Ambient Temperature	Maximum Clock Frequency	
		2MHz	3MHz
Hermetic DIP	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	Am9511ADC	Am9511A-1DC
	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	Am9511ADM	Am9511A-1DM

INTERFACE SIGNAL DESCRIPTION**VCC:** +5V Power Supply**VDD:** +12V Power Supply**VSS:** Ground**CLK (Clock, Input)**

An external timing source connected to the CLK input provides the necessary clocking. The CLK input can be asynchronous to the RD and WR control signals.

RESET (Reset, Input)

A HIGH on this input causes initialization. Reset terminates any operation in progress, and clears the status register to zero. The internal stack pointer is initialized and the contents of the stack may be affected but the command register is not affected by the reset operation. After a reset the $\overline{\text{END}}$ output will be HIGH, and the SVREQ output will be LOW. For proper initialization, the RESET input must be HIGH for at least five CLK periods following stable power supply voltages and stable clock.

C/D (Command/Data Select, Input)

The C/D input together with the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ inputs determines the type of transfer to be performed on the data bus as follows:

C/D	$\overline{\text{RD}}$	$\overline{\text{WR}}$	Function
L	H	L	Push data byte into the stack
L	L	H	Pop data byte from the stack
H	H	L	Enter command byte from the data bus
H	L	H	Read Status
X	L	L	Undefined

L = LOW

H = HIGH

X = DON'T CARE

END (End of Execution, Output)

A LOW on this output indicates that execution of the current command is complete. This output will be cleared HIGH by activating the EACK input LOW or performing any read or write operation or device initialization using the RESET. If EACK is tied LOW, the END output will be a pulse (see EACK description). This is an open drain output and requires a pull up to +5V.

Reading the status register while a command execution is in progress is allowed. However any read or write operation clears the flip-flop that generates the $\overline{\text{END}}$ output. Thus such continuous reading could conflict with internal logic setting the $\overline{\text{END}}$ flip-flop at the completion of command execution.

EACK (End Acknowledge, Output)

This input when LOW makes the $\overline{\text{END}}$ output go LOW. As mentioned earlier HIGH on the $\overline{\text{END}}$ output signals completion of a command execution. The $\overline{\text{END}}$ output signal is derived from an internal flip-flop which is clocked at the completion of a command. This flip-flop is clocked to the reset state when EACK is LOW. Consequently, if the EACK is tied LOW, the $\overline{\text{END}}$ output will be a pulse that is approximately one CLK period wide.

SVREQ (Service Request, Output)

A HIGH on this output indicates completion of a command. In this sense this output is same as the $\overline{\text{END}}$ output. However, whether the SVREQ output will go HIGH at the completion of a command or not is determined by a service request bit in the command register. This bit must be 1 for SVREQ to go HIGH. The SVREQ can be cleared (i.e., go LOW) by activating the SVACK input LOW or initializing the device using the RESET.

Also, the SVREQ will be automatically cleared after completion of any command that has the service request bit as 0.

SVACK (Service Acknowledge, Input)

A LOW on this input activates the reset input of the flip-flop generating the SVREQ output. If the SVACK input is permanently tied LOW, it will conflict with the internal setting of the flip-flop to generate the SVREQ output. Thus the SVREQ indication cannot be relied upon if the SVACK is tied LOW.

DB0-DB7 (Bidirectional Data Bus, Input/Output)

These eight bidirectional lines are used to transfer command, status and operand information between the device and the host processor. DB0 is the least significant and DB7 is the most significant bit position. HIGH on the data bus line corresponds to 1 and LOW corresponds to 0.

When pushing operands on the stack using the data bus, the least significant byte must be pushed first and most significant byte last. When popping the stack to read the result of an operation, the most significant byte will be available on the data bus first and the least significant byte will be the last. Moreover, for pushing operands and popping results, the number of transactions must be equal to the proper number of bytes appropriate for the chosen format. Otherwise, the internal byte pointer will not be aligned properly. The Am9511A single precision format requires 2 bytes, double precision and floating-point formats require 4 bytes.

CS (Chip Select, Input)

This input must be LOW to accomplish any read or write operation to the Am9511A.

To perform a write operation data is presented on DB0 through DB7 lines, C/D is driven to an appropriate level and the $\overline{\text{CS}}$ input is made LOW. However, actual writing into the Am9511A cannot start until $\overline{\text{WR}}$ is made LOW. After initiating the write operation by a $\overline{\text{WR}}$ HIGH to LOW transition, the $\overline{\text{PAUSE}}$ output will go LOW momentarily (TPPWW).

The $\overline{\text{WR}}$ input can go HIGH after $\overline{\text{PAUSE}}$ goes HIGH. The data lines, C/D input and the $\overline{\text{CS}}$ input can change when appropriate hold time requirements are satisfied. See write timing diagram for details.

To perform a read operation an appropriate logic level is established on the C/D input and $\overline{\text{CS}}$ is made LOW. The Read operation does not start until the $\overline{\text{RD}}$ input goes LOW. $\overline{\text{PAUSE}}$ will go LOW for a period of TPPWR. When $\overline{\text{PAUSE}}$ goes back HIGH again, it indicates that read operation is complete and the required information is available on the DB0 through DB7 lines. This information will remain on the data lines as long as $\overline{\text{RD}}$ input is LOW. The $\overline{\text{RD}}$ input can return HIGH anytime after $\overline{\text{PAUSE}}$ goes HIGH. The $\overline{\text{CS}}$ input and C/D inputs can change anytime after $\overline{\text{RD}}$ returns HIGH. See read timing diagram for details.

 $\overline{\text{RD}}$ (Read, Input)

A LOW on this input is used to read information from an internal location and gate that information on to the data bus. The $\overline{\text{CS}}$ input must be LOW to accomplish the read operation. The C/D input determines what internal location is of interest. See C/D, $\overline{\text{CS}}$ input descriptions and read timing diagram for details. If the $\overline{\text{END}}$ output was LOW, performing any read operation will make the $\overline{\text{END}}$ output go HIGH after the HIGH to LOW transition of the $\overline{\text{RD}}$ input (assuming $\overline{\text{CS}}$ is LOW).

WR (Write, Input)

A LOW on this input is used to transfer information from the data bus into an internal location. The \overline{CS} must be LOW to accomplish the write operation. The C/\overline{D} determines which internal location is to be written. See C/\overline{D} , \overline{CS} input descriptions and write timing diagram for details.

If the \overline{END} output was LOW, performing any write operation will make the \overline{END} output go HIGH after the LOW to HIGH transition of the \overline{WR} input (assuming \overline{CS} is LOW).

PAUSE (Pause, Output)

This output is a handshake signal used while performing read or write transactions with the Am9511A. A LOW at this output indicates that the Am9511A has not yet completed its information transfer with the host over the data bus. During a read operation, after \overline{CS} went LOW, the \overline{PAUSE} will become LOW shortly (TRP) after \overline{RD} goes LOW. \overline{PAUSE} will return high only after the data bus contains valid output data. The \overline{CS} and \overline{RD} should remain LOW when \overline{PAUSE} is LOW. The \overline{RD} may go high anytime after \overline{PAUSE} goes HIGH. During a write operation, after \overline{CS} went LOW, the \overline{PAUSE} will be LOW for a very short duration (TPPW) after \overline{WR} goes LOW. Since the minimum of TPPW is 0, the \overline{PAUSE} may not go LOW at all for fast devices. \overline{WR} may go HIGH anytime after \overline{PAUSE} goes HIGH.

FUNCTIONAL DESCRIPTION

Major functional units of the Am9511A are shown in the block diagram. The Am9511A employs a microprogram controlled stack oriented architecture with 16-bit wide data paths.

The Arithmetic Logic Unit (ALU) receives one of its operands from the Operand Stack. This stack is an 8-word by 16-bit 2-port memory with last in-first out (LIFO) attributes. The second operand to the ALU is supplied by the internal 16-bit bus. In addition to supplying the second operand, this bidirectional bus also carries the results from the output of the ALU when required. Writing into the Operand Stack takes place from this internal 16-bit bus when required. Also connected to this bus are the Constant ROM and Working Registers. The ROM provides the required constants to perform the mathematical operations (Chebyshev Algorithms) while the Working Registers provide storage for the intermediate values during command execution.

Communication between the external world and the Am9511A takes place on eight bidirectional input/output lines DB0 through DB7 (Data Bus). These signals are gated to the internal eight-bit

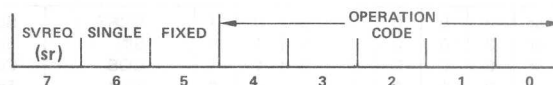
bus through appropriate interface and buffer circuitry. Multiplexing facilities exist for bidirectional communication between the internal eight and sixteen-bit buses. The Status Register and Command Register are also accessible via the eight-bit bus.

The Am9511A operations are controlled by the microprogram contained in the Control ROM. The Program Counter supplies the microprogram addresses and can be partially loaded from the Command Register. Associated with the Program Counter is the Subroutine Stack where return addresses are held during subroutine calls in the microprogram. The Microinstruction Register holds the current microinstruction being executed. This register facilitates pipelined microprogram execution. The Instruction Decode logic generates various internal control signals needed for the Am9511A operation.

The Interface Control logic receives several external inputs and provides handshake related outputs to facilitate interfacing the Am9511A to microprocessors.

COMMAND FORMAT

Each command entered into the Am9511A consists of a single 8-bit byte having the format illustrated below:



Bits 0-4 select the operation to be performed as shown in the table. Bits 5-6 select the data format for the operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated on by fixed point commands (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are indicated; if bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (SVREQ) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin (SVACK) or until completion of execution of a succeeding command where bit 7 is 0. Each command issued to the Am9511A requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains low.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description
7	6	5	4	3	2	1	0		
FIXED-POINT 16-BIT									
sr	1	1	0	1	1	0	0	SADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	0	1	SSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	1	0	SMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	1	1	1	0	1	1	0	SMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	1	1	0	1	1	1	1	SDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FIXED-POINT 32-BIT									
sr	0	1	0	1	1	0	0	DADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	0	1	DSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	1	0	DMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	0	1	1	0	1	1	0	DMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	0	1	0	1	1	1	1	DDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FLOATING-POINT 32-BIT									
sr	0	0	1	0	0	0	0	FADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	0	1	FSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	0	FMUL	Multiply NOS by TOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	1	FDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
DERIVED FLOATING-POINT FUNCTIONS									
sr	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
sr	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
sr	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS.
sr	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
sr	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
sr	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
sr	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
sr	0	0	0	1	0	1	0	EXP	Exponential (e ^x) of TOS. Result in TOS.
sr	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result in NOS. Pop Stack.
DATA MANIPULATION COMMANDS									
sr	0	0	0	0	0	0	0	NOP	No Operation
sr	0	0	1	1	1	1	1	FIXS	Convert TOS from floating point to 16-bit fixed point format.
sr	0	0	1	1	1	1	0	FIXD	Convert TOS from floating point to 32-bit fixed point format.
sr	0	0	1	1	1	0	1	FLTS	Convert TOS from 16-bit fixed point to floating point format.
sr	0	0	1	1	1	0	0	FLTD	Convert TOS from 32-bit fixed point to floating point format.
sr	1	1	1	0	1	0	0	CHSS	Change sign of 16-bit fixed point operand on TOS.
sr	0	1	1	0	1	0	0	CHSD	Change sign of 32-bit fixed point operand on TOS.
sr	0	0	1	0	1	0	1	CHSF	Change sign of floating point operand on TOS.
sr	1	1	1	0	1	1	1	PTOS	Push 16-bit fixed point operand on TOS to NOS (Copy)
sr	0	1	1	0	1	1	1	PTOD	Push 32-bit fixed point operand on TOS to NOS. (Copy)
sr	0	0	1	0	1	1	1	PTOF	Push floating point operand on TOS to NOS. (Copy)
sr	1	1	1	1	0	0	0	POPS	Pop 16-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	1	1	1	0	0	0	POPD	Pop 32-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	0	1	1	0	0	0	POPF	Pop floating point operand from TOS. NOS becomes TOS.
sr	1	1	1	1	0	0	1	XCHS	Exchange 16-bit fixed point operands TOS and NOS.
sr	0	1	1	1	0	0	1	XCHD	Exchange 32-bit fixed point operands TOS and NOS.
sr	0	0	1	1	0	0	1	XCHF	Exchange floating point operands TOS and NOS.
sr	0	0	1	1	0	1	0	PUPI	Push floating point constant "π" onto TOS. Previous TOS becomes NOS.

NOTES:

1. TOS means Top of Stack. NOS means Next on Stack.
2. AMD Application Brief "Algorithm Details for the Am9511A APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
4. The trigonometric functions handle angles in radians, not degrees.
5. No remainder is available for the fixed-point divide functions.
6. Results will be undefined for any combination of command coding bits not specified in this table.

COMMAND INITIATION

After properly positioning the required operands on the stack, a command may be issued. The procedure for initiating a command execution is as follows:

1. Enter the appropriate command on the DB0-DB7 lines.
2. Establish HIGH on the C/\overline{D} input.
3. Establish LOW on the \overline{CS} input.
4. Establish LOW on the \overline{WR} input after an appropriate set up time (see timing diagrams).
5. Sometime after the HIGH to LOW level transition of \overline{WR} input, the \overline{PAUSE} output will become LOW. After a delay of $TPPW$, it will go HIGH to acknowledge the write operation. The \overline{WR} input can return to HIGH anytime after \overline{PAUSE} going HIGH. The DB0-DB7, C/\overline{D} and \overline{CS} inputs are allowed to change after the hold time requirements are satisfied (see timing diagram).

An attempt to issue a new command while the current command execution is in progress is allowed. Under these circumstances, the \overline{PAUSE} output will not go HIGH until the current command execution is completed.

OPERAND ENTRY

The Am9511A commands operate on the operands located at the TOS and NOS and results are returned to the stack at NOS and then popped to TOS. The operands required for the Am9511A are one of three formats — single precision fixed-point (2 bytes), double precision fixed-point (4 bytes) or floating-point (4 bytes). The result of an operation has the same format as the operands except for float to fix or fix to float commands.

Operands are always entered into the stack least significant byte first and most significant byte last. The following procedure must be followed to enter operands onto the stack:

1. The lower significant operand byte is established on the DB0-DB7 lines.
2. A LOW is established on the C/\overline{D} input to specify that data is to be entered into the stack.
3. The \overline{CS} input is made LOW.
4. After appropriate set up time (see timing diagrams), the \overline{WR} input is made LOW. The \overline{PAUSE} output will become LOW.
5. Sometime after this event, the \overline{PAUSE} will return HIGH to indicate that the write operation has been acknowledged.
6. Anytime after the \overline{PAUSE} output goes HIGH the \overline{WR} input can be made HIGH. The DB0-DB7, C/\overline{D} and \overline{CS} inputs can change after appropriate hold time requirements are satisfied (see timing diagrams).

The above procedure must be repeated until all bytes of the operand are pushed into the stack. It should be noted that for single precision fixed-point operands 2 bytes should be pushed and 4 bytes must be pushed for double precision fixed-point or floating-point. Not pushing all the bytes of a quantity will result in byte pointer misalignment.

The Am9511A stack can accommodate 8 single precision fixed-point quantities or 4 double precision fixed-point or floating-point quantities. Pushing more quantities than the capacity of the stack will result in loss of data which is usual with any LIFO stack.

DATA REMOVAL

Result from an operation will be available at the TOS. Results can be transferred from the stack to the data bus by reading the stack. When the stack is popped for results, the most significant byte is available first and the least significant byte last. A result is always of the same precision as the operands that produced it

except for format conversion commands. Thus when the result is taken from the stack, the total number of bytes popped out should be appropriate with the precision — single precision results are 2 bytes and double precision and floating-point results are 4 bytes. The following procedure must be used for reading the result from the stack:

1. A LOW is established on the C/\overline{D} input.
2. The \overline{CS} input is made LOW.
3. After appropriate set up time (see timing diagrams), the \overline{RD} input is made LOW. The \overline{PAUSE} will become LOW.
4. Sometime after this, \overline{PAUSE} will return HIGH indicating that the data is available on the DB0-DB7 lines. This data will remain on the DB0-DB7 lines as long as the \overline{RD} input remains LOW.
5. Anytime after \overline{PAUSE} goes HIGH, the \overline{RD} input can return HIGH to complete transaction.
6. The \overline{CS} and C/\overline{D} inputs can change after appropriate hold time requirements are satisfied (see timing diagram).
7. Repeat this procedure until all bytes appropriate for the precision of the result are popped out.

Reading of the stack does not alter its data; it only adjusts the byte pointer. If more data is popped than the capacity of the stack, the internal byte pointer will wrap around and older data will be read again, consistent with the LIFO stack.

STATUS READ

The Am9511A status register can be read without any regard to whether a command is in progress or not. The only implication that has to be considered is the effect this might have on the END output discussed in the signal descriptions.

The following procedure must be followed to accomplish status register reading.

1. Establish HIGH on the C/\overline{D} input.
2. Establish LOW on the \overline{CS} input.
3. After appropriate set up time (see timing diagram) \overline{RD} input is made LOW. The \overline{PAUSE} will become LOW.
4. Sometime after the HIGH to LOW transition of \overline{RD} input, the \overline{PAUSE} will become HIGH indicating that status register contents are available on the DB0-DB7 lines. The status data will remain on DB0-DB7 as long as \overline{RD} input is LOW.
5. The \overline{RD} input can be returned HIGH anytime after \overline{PAUSE} goes HIGH.
6. The C/\overline{D} input and \overline{CS} input can change after satisfying appropriate hold time requirements (see timing diagram).

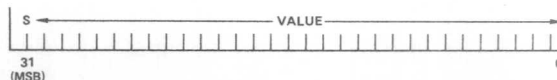
DATA FORMATS

The Am9511A Arithmetic Processing Unit handles operands in both fixed-point and floating-point formats. Fixed-point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

16-BIT FIXED-POINT FORMAT



32-BIT FIXED-POINT FORMAT



The sign (positive or negative) of the operand is located in the most significant bit (MSB). Positive values are represented by a sign bit of zero ($S = 0$). Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 ($S = 1$). The range of values that may be accommodated by each of these formats is $-32,768$ to $+32,767$ for single precision and $-2,147,483,648$ to $+2,147,483,647$ for double precision.

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation.

$$(5.83 \times 10^2)(8.16 \times 10^1) = (4.75728 \times 10^4)$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e.g., 47,572.8). The value-portion of the data is called the mantissa. The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it. The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field. For example, in decimal notation if the exponent field is two digits wide, and the mantissa is five digits, a range of values (positive or negative) from 1.0000×10^{-99} to $9.9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example, since each would be expressed as: 1.2345×10^5 . The sixth digit has been discarded. In most applications where the dynamic range of values to be represented is large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range.

The Am9511 is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

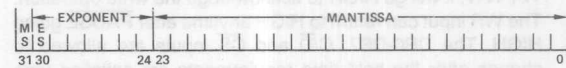
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.11001001×2^7 . The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned} \text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^7 \\ &= (0.5 + 0.25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100.5 \end{aligned}$$

FLOATING POINT FORMAT

The format for floating-point values in the Am9511A is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as an unbiased two's complement 7-bit value having a range of -64 to $+63$. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be to the left of the most significant mantissa bit (bit 23). All floating-point data values must be normalized. Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros.



The range of values that can be represented in this format is $\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero.

STATUS REGISTER

The Am9511A contains an eight bit status register with the following bit assignments:

BUSY	SIGN	ZERO	ERROR CODE				CARRY
7	6	5	4	3	2	1	0

- BUSY:** Indicates that Am9511A is currently executing a command (1 = Busy).
- SIGN:** Indicates that the value on the top of stack is negative (1 = Negative).
- ZERO:** Indicates that the value on the top of stack is zero (1 = Value is zero).
- ERROR CODE:** This field contains an indication of the validity of the result of the last operation. The error codes are:
- 0000 – No error
 - 1000 – Divide by zero
 - 0100 – Square root or log of negative number
 - 1100 – Argument of inverse sine, cosine, or e^x too large
 - XX10 – Underflow
 - XX01 – Overflow
- CARRY:** Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow)

If the BUSY bit in the status register is a one, the other status bits are not defined; if zero, indicating not busy, the operation is complete and the other status bits are defined as given above.

Table 1.

Command Mnemonic	Hex Code (sr = 1)	Hex Code (sr = 0)	Execution Cycles	Summary Description
16-BIT FIXED-POINT OPERATIONS				
SADD	EC	6C	16-18	Add TOS to NOS. Result to NOS. Pop Stack.
SSUB	ED	6D	30-32	Subtract TOS from NOS. Result to NOS. Pop Stack.
SMUL	EE	6E	84-94	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
SMUU	F6	76	80-98	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
SDIV	EF	6F	84-94	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FIXED-POINT OPERATIONS				
DADD	AC	2C	20-22	Add TOS to NOS. Result to NOS. Pop Stack.
DSUB	AD	2D	38-40	Subtract TOS from NOS. Result to NOS. Pop Stack.
DMUL	AE	2E	194-210	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
DMUU	B6	36	182-218	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
DDIV	AF	2F	196-210	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT PRIMARY OPERATIONS				
FADD	90	10	54-368	Add TOS to NOS. Result to NOS. Pop Stack.
FSUB	91	11	70-370	Subtract TOS from NOS. Result to NOS. Pop Stack.
FMUL	92	12	146-168	Multiply NOS by TOS. Result to NOS. Pop Stack.
FDIV	93	13	154-184	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT DERIVED OPERATIONS				
SQRT	81	01	782-870	Square Root of TOS. Result to TOS.
SIN	82	02	3796-4808	Sine of TOS. Result to TOS.
COS	83	03	3840-4878	Cosine of TOS. Result to TOS.
TAN	84	04	4894-5886	Tangent of TOS. Result to TOS.
ASIN	85	05	6230-7938	Inverse Sine of TOS. Result to TOS.
ACOS	86	06	6304-8284	Inverse Cosine of TOS. Result to TOS.
ATAN	87	07	4992-6536	Inverse Tangent of TOS. Result to TOS.
LOG	88	08	4474-7132	Common Logarithm of TOS. Result to TOS.
LN	89	09	4298-6956	Natural Logarithm of TOS. Result to TOS.
EXP	8A	0A	3794-4878	e raised to power in TOS. Result to TOS.
PWR	8B	0B	8290-12032	NOS raised to power in TOS. Result to NOS. Pop Stack.
DATA AND STACK MANIPULATION OPERATIONS				
NOP	80	00	4	No Operation. Clear or set SVREQ.
FIXS	9F	1F	90-214	Convert TOS from floating point format to fixed point format.
FIXD	9E	1E	90-336	
FLTS	9D	1D	62-156	
FLTD	9C	1C	56-342	
CHSS	F4	74	22-24	Change sign of fixed point operand on TOS.
CHSD	B4	34	26-28	
CHSF	95	15	16-20	Change sign of floating point operand on TOS.
PTOS	F7	77	16	
PTOD	B7	37	20	Push stack. Duplicate NOS in TOS.
PTOF	97	17	20	
POPS	F8	78	10	Pop stack. Old NOS becomes new TOS. Old TOS rotates to bottom.
POPD	B8	38	12	
POPF	98	18	12	
XCHS	F9	79	18	
XCHD	B9	39	26	Exchange TOS and NOS.
XCHF	99	19	26	
PUPI	9A	1A	16	Push floating point constant π onto TOS. Previous TOS becomes NOS.

COMMAND DESCRIPTIONS

This section contains detailed descriptions of the APU commands. They are arranged in alphabetical order by command mnemonic. In the descriptions, TOS means Top Of Stack and NOS means Next On Stack.

All derived functions except Square Root use Chebyshev polynomial approximating algorithms. This approach is used to help minimize the internal microprogram, to minimize the maximum error values and to provide a relatively even distribution of errors over the data range. The basic arithmetic operations are used by the derived functions to compute the various Chebyshev terms. The basic operations may produce error codes in the status register as a result.

Execution times are listed in terms of clock cycles and may be converted into time values by multiplying by the clock period used. For example, an execution time of 44 clock cy-

cles when running at a 3MHz rate translates to 14 microseconds ($44 \times 32\mu s = 14\mu s$). Variations in execution cycles reflect the data dependency of the algorithms.

In some operations exponent overflow or underflow may be possible. When this occurs, the exponent returned in the result will be 128 greater or smaller than its true value.

Many of the functions use portions of the data stack as scratch storage during development of the results. Thus previous values in those stack locations will be lost. Scratch locations destroyed are listed in the command descriptions and shown with the crossed-out locations in the Stack Contents After diagram.

Table 1 is a summary of all the Am9511A commands. It shows the hex codes for each command, the mnemonic abbreviation, a brief description and the execution time in clock cycles. The commands are grouped by functional classes.

The command mnemonics in alphabetical order are shown below in Table 2.

Table 2.
Command Mnemonics in Alphabetical Order.

ACOS	ARCCOSINE	LOG	COMMON LOGARITHM
ASIN	ARCSINE	LN	NATURAL LOGARITHM
ATAN	ARCTANGENT	NOP	NO OPERATION
CHSD	CHANGE SIGN DOUBLE	POPD	POP STACK DOUBLE
CHSF	CHANGE SIGN FLOATING	POPF	POP STACK FLOATING
CHSS	CHANGE SIGN SINGLE	POPS	POP STACK SINGLE
COS	COSINE	PTOD	PUSH STACK DOUBLE
DADD	DOUBLE ADD	PTOF	PUSH STACK FLOATING
DDIV	DOUBLE DIVIDE	PTOS	PUSH STACK SINGLE
DMUL	DOUBLE MULTIPLY LOWER	PUPI	PUSH π
DMUU	DOUBLE MULTIPLY UPPER	PWR	POWER (x^y)
DSUB	DOUBLE SUBTRACT	SADD	SINGLE ADD
EXP	EXPONENTIATION (e^x)	SDIV	SINGLE DIVIDE
FADD	FLOATING ADD	SIN	SINE
FDIV	FLOATING DIVIDE	SMUL	SINGLE MULTIPLY LOWER
FIXD	FIX DOUBLE	SMUU	SINGLE MULTIPLY UPPER
FIXS	FIX SINGLE	SQRT	SQUARE ROOT
FLTD	FLOAT DOUBLE	SSUB	SINGLE SUBTRACT
FLTS	FLOAT SINGLE	TAN	TANGENT
FMUL	FLOATING MULTIPLY	XCHD	EXCHANGE OPERANDS DOUBLE
FSUB	FLOATING SUBTRACT	XCHF	EXCHANGE OPERANDS FLOATING
		XCHS	EXCHANGE OPERANDS SINGLE

ACOS

32-BIT FLOATING-POINT INVERSE COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	0

Hex Coding: 86 with sr = 1
06 with sr = 0

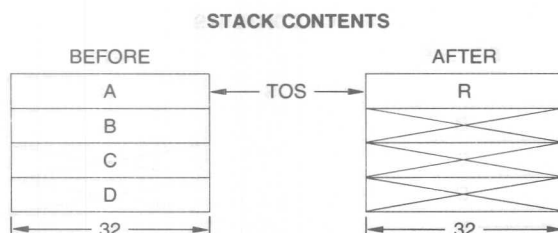
Execution Time: 6304 to 8284 clock cycles

Description:

The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse cosine of A. The result R is a value in radians between 0 and π . Initial operands A, B, C and D are lost. ACOS will accept all input data values within the range of -1.0 to $+1.0$. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ACOS exhibits a maximum relative error of 2.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field



ASIN

32-BIT FLOATING-POINT INVERSE SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	1

Hex Coding: 85 with sr = 1
05 with sr = 0

Execution Time: 6230 to 7938 clock cycles

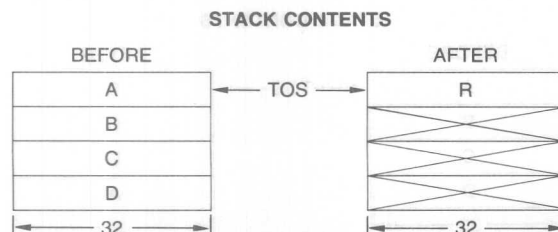
Description:

The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse sine of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, B, C and D are lost.

ASIN will accept all input data values within the range of -1.0 to $+1.0$. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ASIN exhibits a maximum relative error of 4.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field



ATAN

32-BIT FLOATING-POINT INVERSE TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	1

Hex Coding: 87 with sr = 1
07 with sr = 0

Execution Time: 4992 to 6536 clock cycles

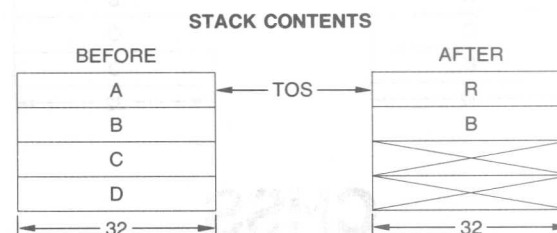
Description:

The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse tangent of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, C and D are lost. Operand B is unchanged.

ATAN will accept all input data values that can be represented in the floating point format.

Accuracy: ATAN exhibits a maximum relative error of 3.0×10^{-7} over the input data range.

Status Affected: Sign, Zero



CHSD

32-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	0	0

Hex Coding: B4 with sr = 1
34 with sr = 0

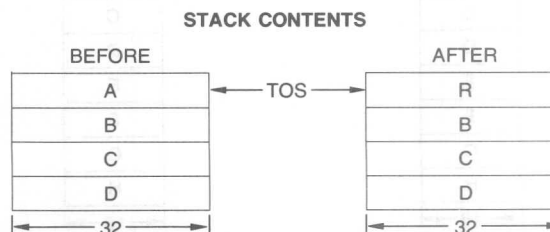
Execution Time: 26 to 28 clock cycles

Description:

The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. Other entries in the stack are not disturbed.

Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Error Field (overflow)



CHSF

32-BIT FLOATING-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	0	1

Hex Coding: 95 with sr = 1
15 with sr = 0

Execution Time: 16 to 20 clock cycles

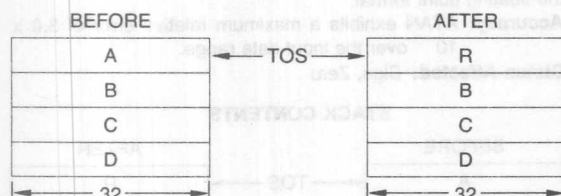
Description:

The sign of the mantissa of the 32-bit floating-point operand A at the TOS is inverted. The result R replaces A at the TOS. Other stack entries are unchanged.

If A is input as zero (mantissa MSB = 0), no change is made.

Status Affected: Sign, Zero

STACK CONTENTS



CHSS

16-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	0	0

Hex Coding: F4 with sr = 1
74 with sr = 0

Execution Time: 22 to 24 clock cycles

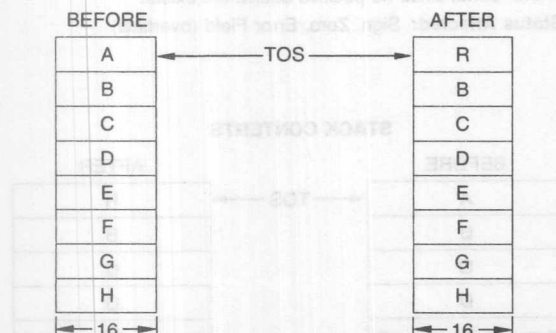
Description:

16-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. All other operands are unchanged.

Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



COS

32-BIT FLOATING-POINT COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	1

Hex Coding: 83 with sr = 1
03 with sr = 0

Execution Time: 3840 to 4878 clock cycles

Description:

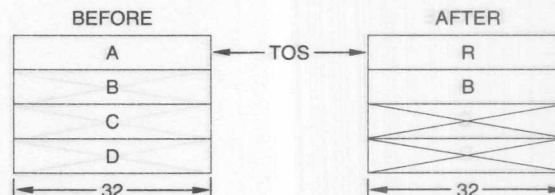
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point cosine of A. A is assumed to be in radians. Operands A, C and D are lost. B is unchanged.

The COS function can accept any input data value that can be represented in the data format. All input values are range reduced to fall within an interval of $-\pi/2$ to $+\pi/2$ radians.

Accuracy: COS exhibits a maximum relative error of 5.0×10^{-7} for all input data values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



DADD

32-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	0

Hex Coding: AC with sr = 1
2C with sr = 0

Execution Time: 20 to 22 clock cycles

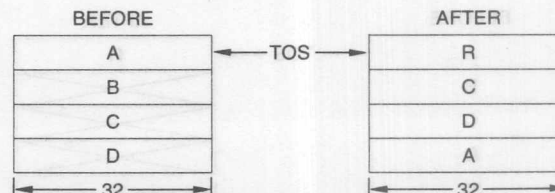
Description:

The 32-bit fixed-point two's complement integer operand A at the TOS is added to the 32-bit fixed-point two's complement integer operand B at the NOS. The result R replaces operand B and the Stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged. If the addition generates a carry it is reported in the status register.

If the result is too large to be represented by the data format, the least significant 32 bits of the result are returned and overflow status is reported.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



DDIV

32-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	1

Hex Coding: AF with sr = 1

2F with sr = 0

Execution Time: 196 to 210 clock cycles when A \neq 0

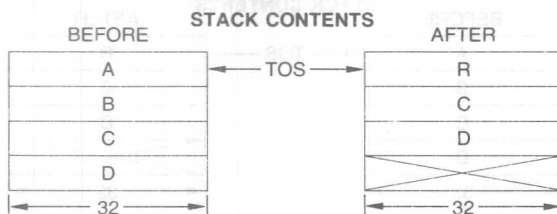
18 clock cycles when A = 0.

Description:

The 32-bit fixed-point two's complement integer operand B at NOS is divided by the 32-bit fixed-point two's complement integer operand A at the TOS. The 32-bit integer quotient R replaces B and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. Operands C and D are unchanged.

If A is zero, R is set equal to B and the divide-by-zero error status will be reported. If either A or B is the most negative value possible in the format, R will be meaningless and the overflow error status will be reported.

Status Affected: Sign, Zero, Error Field



DMUL

32-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	0

Hex Coding: AE with sr = 1

2E with sr = 0

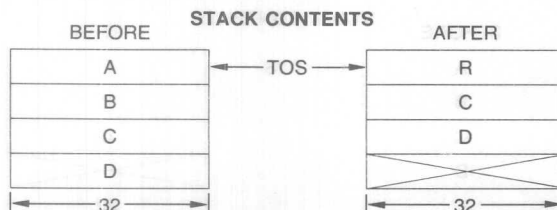
Execution Time: 194 to 210 clock cycles

Description:

The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit least significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.

The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Overflow



DMUU

32-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	0

Hex Coding: B6 with sr = 1

36 with sr = 0

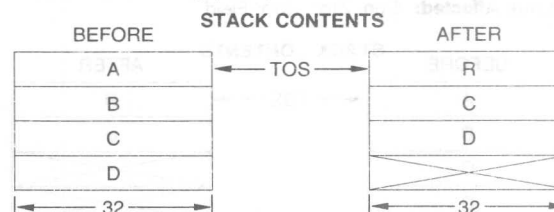
Execution Time: 182 to 218 clock cycles

Description:

The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.

If A or B was the most negative value possible in the format, overflow status is set and R is meaningless.

Status Affected: Sign, Zero, Overflow



DSUB

32-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	1

Hex Coding: AD with sr = 1

2D with sr = 0

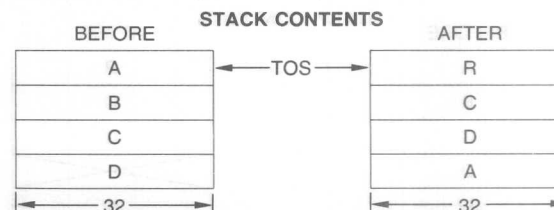
Execution Time: 38 to 40 clock cycles

Description:

The 32-bit fixed-point two's complement operand A at the TOS is subtracted from the 32-bit fixed-point two's complement operand B at the NOS. The difference R replaces B and the stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged.

If the subtraction generates a borrow it is reported in the carry status bit. If A is the most negative value that can be represented in the format the overflow status is set. If the result cannot be represented in the data format range, the overflow bit is set and the 32 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Overflow



EXP

32-BIT FLOATING-POINT e^x

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	0

Hex Coding: 8A with sr = 1
0A with sr = 0

Execution Time: 3794 to 4878 clock cycles for $|A| \leq 1.0 \times 2^5$
34 clock cycles for $|A| > 1.0 \times 2^5$

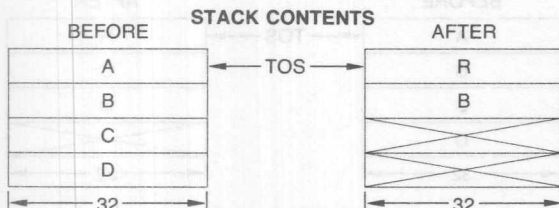
Description:

The base of natural logarithms, e , is raised to an exponent value specified by the 32-bit floating-point operand A at the TOS. The result R of e^A replaces A. Operands A, C and D are lost. Operand B is unchanged.

EXP accepts all input data values within the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$. Input values outside this range will return a code of 1100 in the error field of the status register.

Accuracy: EXP exhibits a maximum relative error of 5.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field



FADD

32-BIT FLOATING-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	0

Hex Coding: 90 with sr = 1
10 with sr = 0

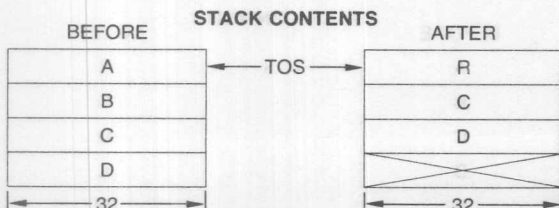
Execution Time: 54 to 368 clock cycles for $A \neq 0$
24 clock cycles for $A = 0$

Description:

32-bit floating-point operand A at the TOS is added to 32-bit floating-point operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

Exponent alignment before the addition and normalization of the result accounts for the variation in execution time. Exponent overflow and underflow are reported in the status register, in which case the mantissa is correct and the exponent is offset by 128.

Status Affected: Sign, Zero, Error Field



FDIV

32-BIT FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	1

Hex Coding: 93 with sr = 1
13 with sr = 0

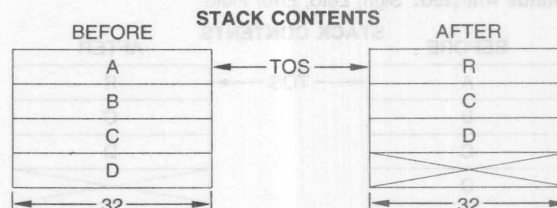
Execution Time: 154 to 184 clock cycles for $A \neq 0$
22 clock cycles for $A = 0$

Description:

32-bit floating-point operand B at NOS is divided by 32-bit floating-point operand A at the TOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

If operand A is zero, R is set equal to B and the divide-by-zero error is reported in the status register. Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field



FIXD

32-BIT FLOATING-POINT TO 32-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	0

Hex Coding: 9E with sr = 1
1E with sr = 0

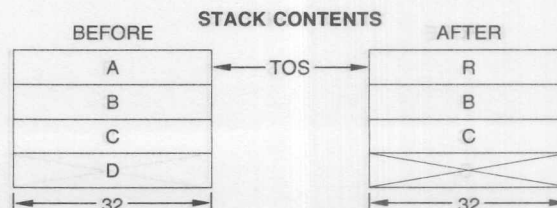
Execution Time: 90 to 336 clock cycles

Description:

32-bit floating-point operand A at the TOS is converted to a 32-bit fixed-point two's complement integer. The result R replaces A. Operands A and D are lost. Operands B and C are unchanged.

If the integer portion of A is larger than 31 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero Overflow



FIXS

32-BIT FLOATING-POINT TO 16-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	1

Hex Coding: 9F with sr = 1
1F with sr = 0

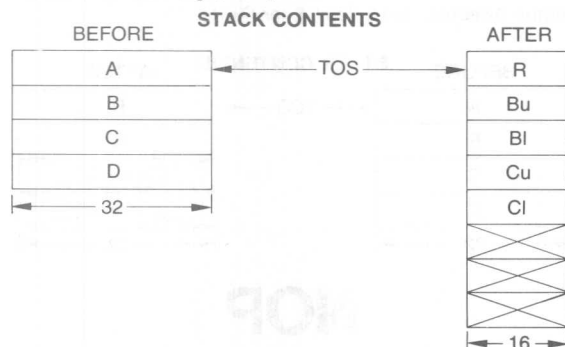
Execution Time: 90 to 214 clock cycles

Description:

32-bit floating-point operand A at the TOS is converted to a 16-bit fixed-point two's complement integer. The result R replaces the lower half of A and the stack is moved up by two bytes so that R occupies the TOS. Operands A and D are lost. Operands B and C are unchanged, but appear as upper (u) and lower (l) halves on the 16-bit wide stack if they are 32-bit operands.

If the integer portion of A is larger than 15 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero, Overflow



FLTD

32-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	0

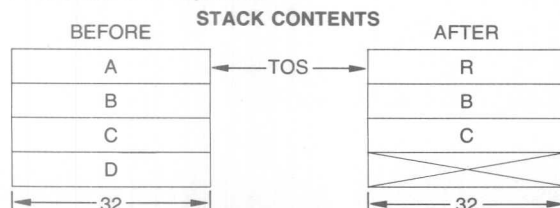
Hex Coding: 9C with sr = 1
1C with sr = 0

Execution Time: 56 to 342 clock cycles

Description:

32-bit fixed-point two's complement integer operand A at the TOS is converted to a 32-bit floating-point number. The result R replaces A at the TOS. Operands A and D are lost. Operands B and C are unchanged.

Status Affected: Sign, Zero



FLTS

16-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	1

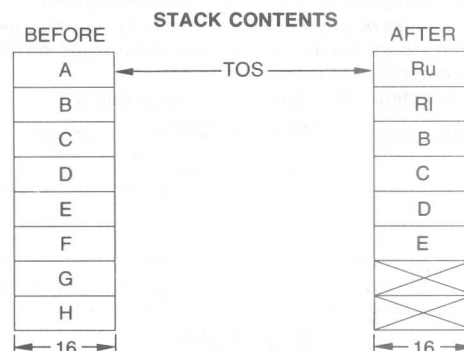
Hex Coding: 9D with sr = 1
1D with sr = 0

Execution Time: 62 to 156 clock cycles

Description:

16-bit fixed-point two's complement integer A at the TOS is converted to a 32-bit floating-point number. The lower half of the result R (Rl) replaces A, the upper half (Ru) replaces H and the stack is moved down so that Ru occupies the TOS. Operands A, F, G and H are lost. Operands B, C, D and E are unchanged.

Status Affected: Sign, Zero



FMUL

32-BIT FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	0

Hex Coding: 92 with sr = 1
12 with sr = 0

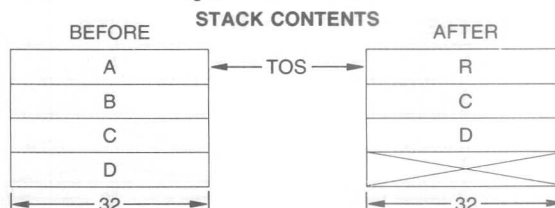
Execution Time: 146 to 168 clock cycles

Description:

32-bit floating-point operand A at the TOS is multiplied by the 32-bit floating-point operand B at the NOS. The normalized result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field



FSUB

32-BIT FLOATING-POINT SUBTRACTION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	1

Hex Coding: 91 with sr = 1
11 with sr = 0

Execution Time: 70 to 370 clock cycles for $A \neq 0$
26 clock cycles for $A = 0$

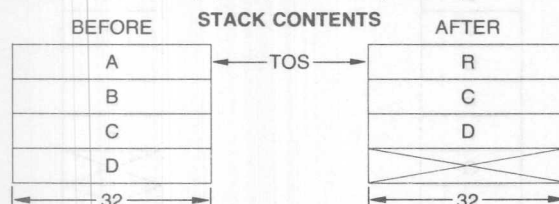
Description:

32-bit floating-point operand A at the TOS is subtracted from 32-bit floating-point operand B at the NOS. The normalized difference R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

Exponent alignment before the subtraction and normalization of the result account for the variation in execution time.

Exponent overflow or underflow is reported in the status register in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field (overflow)



LOG

32-BIT FLOATING-POINT COMMON LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	0

Hex Coding: 88 with sr = 1
08 with sr = 0

Execution Time: 4474 to 7132 clock cycles for $A > 0$
20 clock cycles for $A \leq 0$

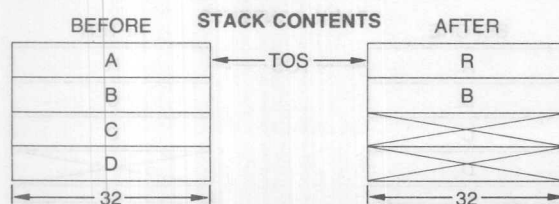
Description:

The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point common logarithm (base 10) of A. Operands A, C and D are lost. Operand B is unchanged.

The LOG function accepts any positive input data value that can be represented by the data format. If LOG of a non-positive value is attempted an error status of 0100 is returned.

Accuracy: LOG exhibits a maximum absolute error of 2.0×10^{-7} for the input range from 0.1 to 10, and a maximum relative error of 2.0×10^{-7} for positive values less than 0.1 or greater than 10.

Status Affected: Sign, Zero, Error Field



LN

32-BIT FLOATING-POINT NATURAL LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	1

Hex Coding: 89 with sr = 1
09 with sr = 0

Execution Time: 4298 to 6956 clock cycles for $A > 0$
20 clock cycles for $A \leq 0$

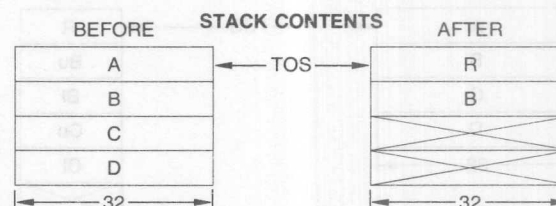
Description:

The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point natural logarithm (base e) of A. Operands A, C and D are lost. Operand B is unchanged.

The LN function accepts all positive input data values that can be represented by the data format. If LN of a non-positive number is attempted an error status of 0100 is returned.

Accuracy: LN exhibits a maximum absolute error of 2×10^{-7} for the input range from e^{-1} to e, and a maximum relative error of 2.0×10^{-7} for positive values less than e^{-1} or greater than e.

Status Affected: Sign, Zero, Error Field



NOP

NO OPERATION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	0

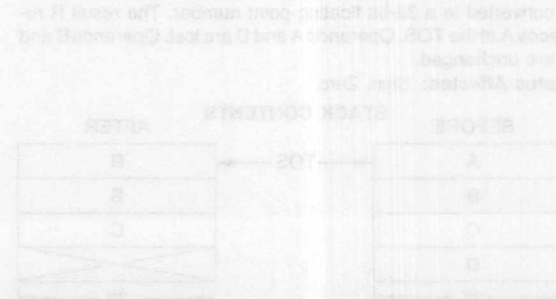
Hex Coding: 80 with sr = 1
00 with sr = 0

Execution Time: 4 clock cycles

Description:

The NOP command performs no internal data manipulations. It may be used to set or clear the service request interface line without changing the contents of the stack.

Status Affected: The status byte is cleared to all zeroes.



POPD

32-BIT
STACK POP

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	1	1	1	0	0	0

Hex Coding: B8 with sr = 1
38 with sr = 0

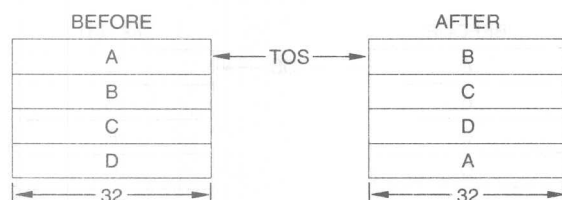
Execution Time: 12 clock cycles

Description:

The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPD and POPF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



POPF

32-BIT
STACK POP

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	1	1	0	0	0

Hex Coding: 98 with sr = 1
18 with sr = 0

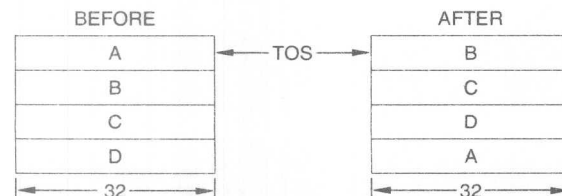
Execution Time: 12 clock cycles

Description:

The 32-bit stack is moved up so that the old NOS becomes the new TOS. The old TOS rotates to the bottom of the stack. All operand values are unchanged. POPF and POPD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



POPS

16-BIT
STACK POP

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	1	0	0	0

Hex Coding: F8 with sr = 1
78 with sr = 0

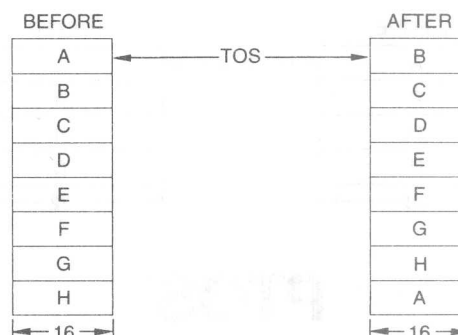
Execution Time: 10 clock cycles

Description:

The 16-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



PTOD

PUSH 32-BIT
TOS ONTO STACK

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	1	1	0	1	1	1

Hex Coding: B7 with sr = 1
37 with sr = 0

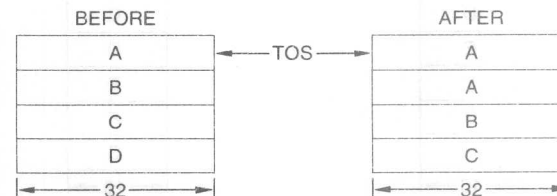
Execution Time: 20 clock cycles

Description:

The 32-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOD and PTOF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



PTOF

PUSH 32-BIT
TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	1	1

Hex Coding: 97 with sr = 1

17 with sr = 0

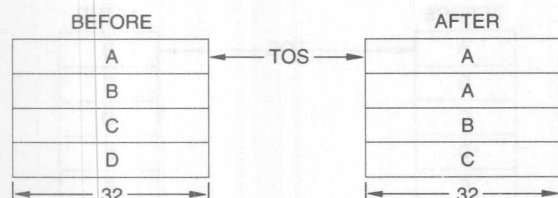
Execution Time: 20 clock cycles

Description:

The 32-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOF and PTOD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



PTOS

PUSH 16-BIT
TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	1	1

Hex Coding: F7 with sr = 1

77 with sr = 0

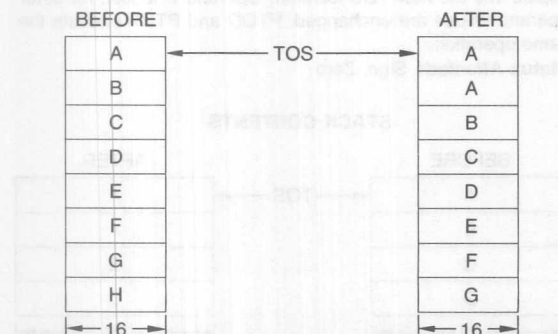
Execution Time: 16 clock cycles

Description:

The 16-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand H is lost and all other operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



PUPI

PUSH 32-BIT
FLOATING-POINT π

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	1	0

Hex Coding: 9A with sr = 1

1A with sr = 0

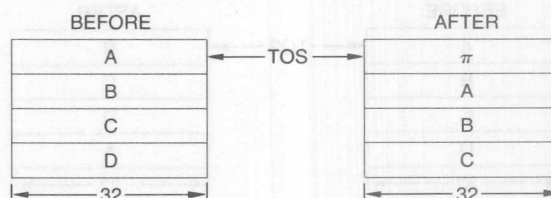
Execution Time: 16 clock cycles

Description:

The 32-bit stack is moved down so that the previous TOS occupies the new NOS location. 32-bit floating-point constant π is entered into the new TOS location. Operand D is lost. Operands A, B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



PWR

32-BIT
FLOATING-POINT X^Y

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	1

Hex Coding: 8B with sr = 1
0B with sr = 0

Execution Time: 8290 to 12032 clock cycles

Description:

32-bit floating-point operand B at the NOS is raised to the power specified by the 32-bit floating-point operand A at the TOS. The result R of B^A replaces B and the stack is moved up so that R occupies the TOS. Operands A, B, and D are lost. Operand C is unchanged.

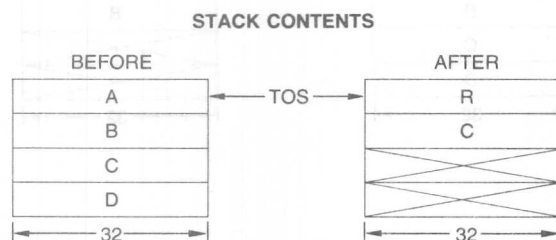
The PWR function accepts all input data values that can be represented in the data format for operand A and all positive values for operand B. If operand B is non-positive an error status of 0100 will be returned. The EXP and LN functions are used to implement PWR using the relationship $B^A = \text{EXP}[A(\text{LN } B)]$. Thus if the term $[A(\text{LN } B)]$ is outside the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$ an error status of 1100 will be returned. Underflow and overflow conditions can occur.

Accuracy: The error performance for PWR is a function of the LN and EXP performance as expressed by:

$$|(\text{Relative Error})_{\text{PWR}}| = |(\text{Relative Error})_{\text{EXP}} + |A|(\text{Absolute Error})_{\text{LN}}|$$

The maximum relative error for PWR occurs when A is at its maximum value while $[A(\text{LN } B)]$ is near 1.0×2^5 and the EXP error is also at its maximum. For most practical applications the relative error for PWR will be less than 7.0×10^{-7} .

Status Affected: Sign, Zero, Error Field



SADD

16-BIT
FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	0

Hex Coding: EC with sr = 1
6C with sr = 0

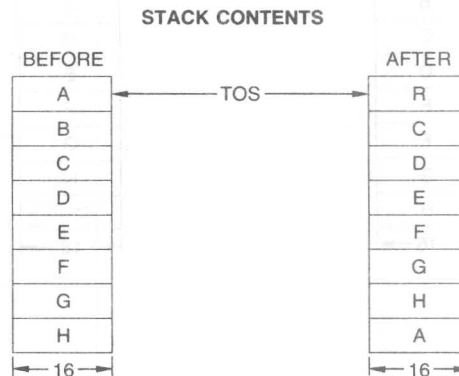
Execution Time: 16 to 18 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is added to 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.

If the addition generates a carry bit it is reported in the status register. If an overflow occurs it is reported in the status register and the 16 least significant bits of the result are returned.

Status Affected: Sign, Zero, Carry, Error Field



SDIV

16-BIT
FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	1

Hex Coding: EF with sr = 1
6F with sr = 0

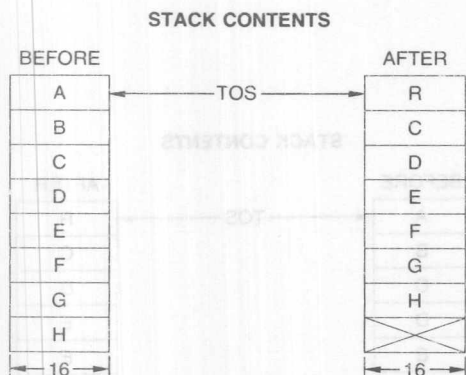
Execution Time: 84 to 94 clock cycles for $A \neq 0$
14 clock cycles for $A = 0$

Description:

16-bit fixed-point two's complement integer operand B at the NOS is divided by 16-bit fixed-point two's complement integer operand A at the TOS. The 16-bit integer quotient R replaces B and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. All other operands are unchanged.

If A is zero, R will be set equal to B and the divide-by-zero error status will be reported.

Status Affected: Sign, Zero, Error Field



SIN

32-BIT
FLOATING-POINT SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	0

Hex Coding: 82 with sr = 1
02 with sr = 0

Execution Time: 3796 to 4808 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

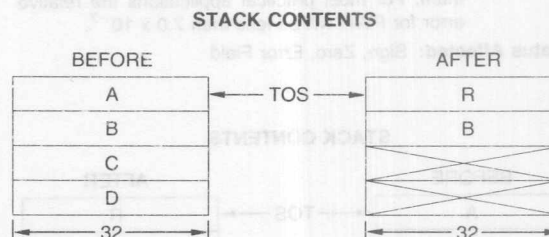
Description:

The SIN function will accept any input data value that can be represented by the data format. All input values are range reduced to fall within the interval $-\pi/2$ to $+\pi/2$ radians. Operands A, C and D are lost. Operand B is unchanged.

The SIN function will accept any input data value that can be represented by the data format. All input values are range reduced to fall within the interval $-\pi/2$ to $+\pi/2$ radians.

Accuracy: SIN exhibits a maximum relative error of 5.0×10^{-7} for input values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero



SMUL

16-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	0

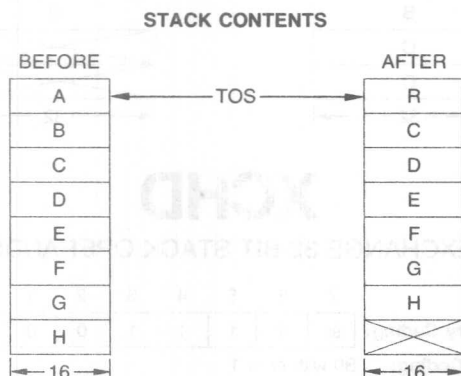
Hex Coding: EE with sr = 1
6E with sr = 0

Execution Time: 84 to 94 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit least significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. All other operands are unchanged. The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field



SMUU

16-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	1	0

Hex Coding: F6 with sr = 1
76 with sr = 0

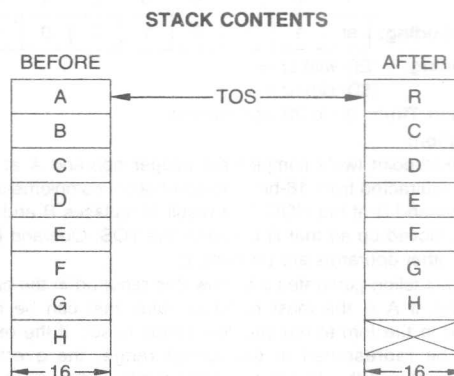
Execution Time: 80 to 98 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. All other operands are unchanged.

If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field



SQRT

32-BIT FLOATING-POINT SQUARE ROOT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	1

Hex Coding: 81 with sr = 1
01 with sr = 0

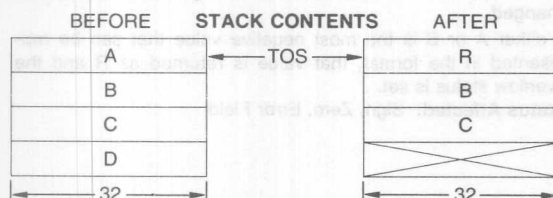
Execution Time: 782 to 870 clock cycles

Description:

32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point square root of A. Operands A and D are lost. Operands B and C are not changed.

SQRT will accept any non-negative input data value that can be represented by the data format. If A is negative an error code of 0100 will be returned in the status register.

Status Affected: Sign, Zero, Error Field



SSUB

16-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	1

Hex Coding: ED with sr = 1
6D with sr = 0

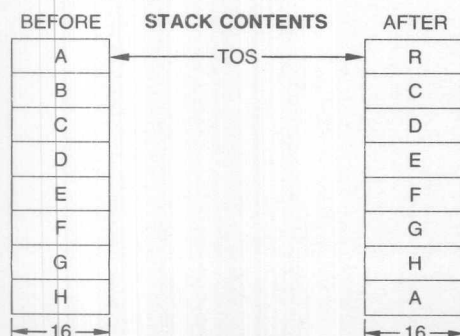
Execution Time: 30 to 32 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is subtracted from 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.

If the subtraction generates a borrow it is reported in the carry status bit. If A is the most negative value that can be represented in the format the overflow status is set. If the result cannot be represented in the format range, the overflow status is set and the 16 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Error Field



TAN

32-BIT FLOATING-POINT TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	0

Hex Coding: 84 with sr = 1
04 with sr = 0

Execution Time: 4894 to 5886 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

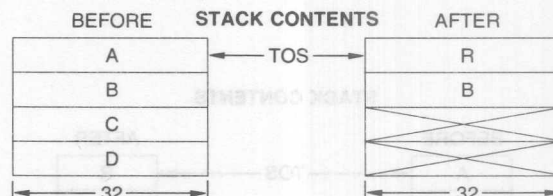
Description:

The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point tangent of A. Operand A is assumed to be in radians. A, C and D are lost. B is unchanged.

The TAN function will accept any input data value that can be represented in the data format. All input data values are range-reduced to fall within $-\pi/4$ to $+\pi/4$ radians. TAN is unbounded for input values near odd multiples of $\pi/2$ and in such cases the overflow bit is set in the status register. For angles smaller than 2^{-12} radians, TAN returns A as the tangent of A.

Accuracy: TAN exhibits a maximum relative error of 5.0×10^{-7} for input data values in the range of -2π to $+2\pi$ radians except for data values near odd multiples of $\pi/2$.

Status Affected: Sign, Zero, Error Field (overflow)



XCHD

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	1

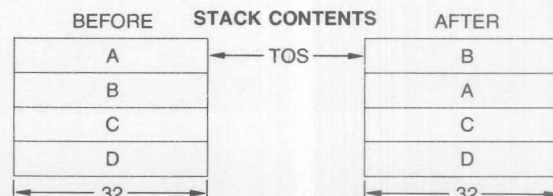
Hex Coding: B9 with sr = 1
39 with sr = 0

Execution Time: 26 clock cycles

Description:

32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero



XCHF

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	0	1

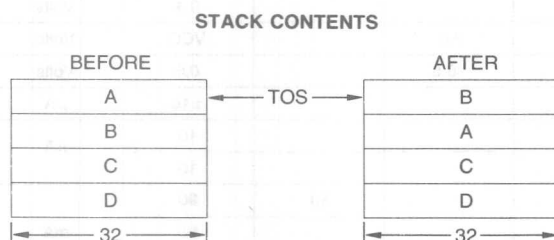
Hex Coding: 99 with sr = 1
19 with sr = 0

Execution Time: 26 clock cycles

Description:

32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero



XCHS

EXCHANGE 16-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	1	0	0	1

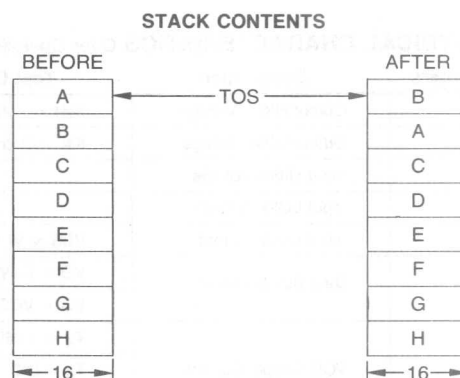
Hex Coding: F9 with sr = 1
79 with sr = 0

Execution Time: 18 clock cycles

Description:

16-bit operand A at the TOS and 16-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operand values are unchanged.

Status Affected: Sign, Zero



Am9511A

MAXIMUM RATINGS beyond which useful life may be impaired

Storage Temperature	-65°C to +150°C
Ambient Temperature Under Bias	-55°C to +125°C
VDD with Respect to VSS	-0.5V to +15.0V
VCC with Respect to VSS	-0.5V to +7.0V
All Signal Voltages with Respect to VSS	-0.5V to +7.0V
Power Dissipation (Package Limitation)	2.0W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VSS	VCC	VDD
Am9511ADC	0°C ≤ T _A ≤ +70°C	0V	+5.0V ±5%	+12V ±5%
Am9511ADM	-55°C ≤ T _A ≤ +125°C	0V	+5.0V ±10%	+12V ±10%

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
VOH	Output HIGH Voltage	IOH = -200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		-0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T _A = +25°C		50	90	mA
		T _A = 0°C			95	
		T _A = -55°C			100	
IDD	VDD Supply Current	T _A = +25°C		50	90	mA
		T _A = 0°C			95	
		T _A = -55°C			100	
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

SWITCHING CHARACTERISTICS over operating range (Notes 2, 3)

Parameters	Description	Am9511A		Am9511A-1		Units
		Min.	Max.	Min.	Max.	
TAPW	$\overline{\text{EACK}}$ LOW Pulse Width	100		75		ns
TCDR	$\text{C}/\overline{\text{D}}$ to $\overline{\text{RD}}$ LOW Set up Time	0		0		ns
TCDW	$\text{C}/\overline{\text{D}}$ to $\overline{\text{WR}}$ LOW Set up Time	0		0		ns
TCPH	Clock Pulse HIGH Width	200		140		ns
TCPL	Clock Pulse LOW Width	240		160		ns
TCSR	$\overline{\text{CS}}$ LOW to $\overline{\text{RD}}$ LOW Set up Time	0		0		ns
TCSW	$\overline{\text{CS}}$ LOW to $\overline{\text{WR}}$ LOW Set up Time	0		0		ns
TCY	Clock Period	480	5000	320	3300	ns
TDW	Data Bus Stable to $\overline{\text{WR}}$ HIGH Set up Time	150		100 (Note 9)		ns
TEAE	$\overline{\text{EACK}}$ LOW to $\overline{\text{END}}$ HIGH Delay		200		175	ns
TEPW	$\overline{\text{END}}$ LOW Pulse Width (Note 4)	400		300		ns
TOP	Data Bus Output Valid to $\overline{\text{PAUSE}}$ HIGH Delay	0		0		ns
TPPWR	$\overline{\text{PAUSE}}$ LOW Pulse Width Read (Note 5)	Data	3.5TCY+50	5.5TCY+300	3.5TCY+50	5.5TCY+200
		Status	1.5TCY+50	3.5TCY+300	1.5TCY+50	3.5TCY+200
TPPWW	$\overline{\text{PAUSE}}$ LOW Pulse Width Write (Note 8)		50		50	ns
TPR	$\overline{\text{PAUSE}}$ HIGH to $\overline{\text{RD}}$ HIGH Hold Time	0		0		ns
TPW	$\overline{\text{PAUSE}}$ HIGH to $\overline{\text{WR}}$ HIGH Hold Time	0		0		ns
TRCD	$\overline{\text{RD}}$ HIGH to $\text{C}/\overline{\text{D}}$ Hold Time	0		0		ns
TRCS	$\overline{\text{RD}}$ HIGH to $\overline{\text{CS}}$ HIGH Hold Time	0		0		ns
TRO	$\overline{\text{RD}}$ LOW to Data Bus ON Delay	50		50		ns
TRP	$\overline{\text{RD}}$ LOW to $\overline{\text{PAUSE}}$ LOW Delay (Note 6)		150		100 (Note 9)	ns
TRZ	$\overline{\text{RD}}$ HIGH to Data Bus OFF Delay	50	200	50	150	ns
TSAPW	$\overline{\text{SVACK}}$ LOW Pulse Width	100		75		ns
TSAR	$\overline{\text{SVACK}}$ LOW to $\overline{\text{SVREQ}}$ LOW Delay		300		200	ns
TWCD	$\overline{\text{WR}}$ HIGH to $\text{C}/\overline{\text{D}}$ Hold Time	60		30		ns
TWCS	$\overline{\text{WR}}$ HIGH to $\overline{\text{CS}}$ HIGH Hold Time	60		30		ns
TWD	$\overline{\text{WR}}$ HIGH to Data Bus Hold Time	20		20		ns
TWI	Write Inactive Time (Note 8)	Command	3TCY	3TCY		ns
		Data	4TCY	4TCY		
TWP	$\overline{\text{WR}}$ LOW to $\overline{\text{PAUSE}}$ LOW Delay (Note 6)		150		100 (Note 9)	ns

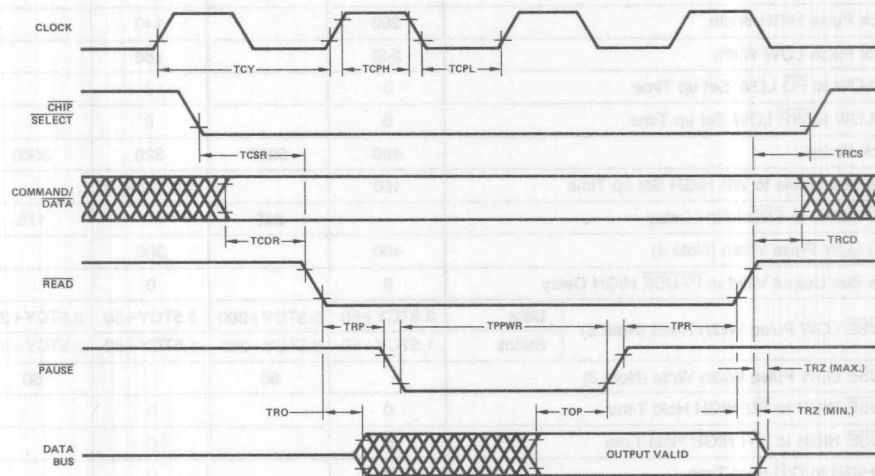
NOTES

- Typical values are for $T_A = 25^\circ\text{C}$, nominal supply voltages and nominal processing parameters.
- Switching parameters are listed in alphabetical order.
- Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF and timing reference levels of 0.8V and 2.0V.
- $\overline{\text{END}}$ low pulse width is specified for $\overline{\text{EACK}}$ tied to VSS. Otherwise TEAE applies.
- Minimum values shown assume no previously entered command is being executed for the data access. If a previously entered command is being executed, $\overline{\text{PAUSE}}$ LOW Pulse Width

- is the time to complete execution plus the time shown. Status may be read at any time without exceeding the time shown.
- $\overline{\text{PAUSE}}$ is pulled low for both command and data operations.
- TEX is the execution time of the current command (see the Command Execution Times table).
- $\overline{\text{PAUSE}}$ low pulse width is less than 50ns when writing into the data port or the control port as long as the duty cycle requirement (TWI) is observed and no previous command is being executed. TWI may be safely violated as long as the extended TPPWW that results is observed. If a previously entered command is being executed, $\overline{\text{PAUSE}}$ LOW Pulse Width is the time to complete execution plus the time shown.
- 150ns for Am9511A-1DM.

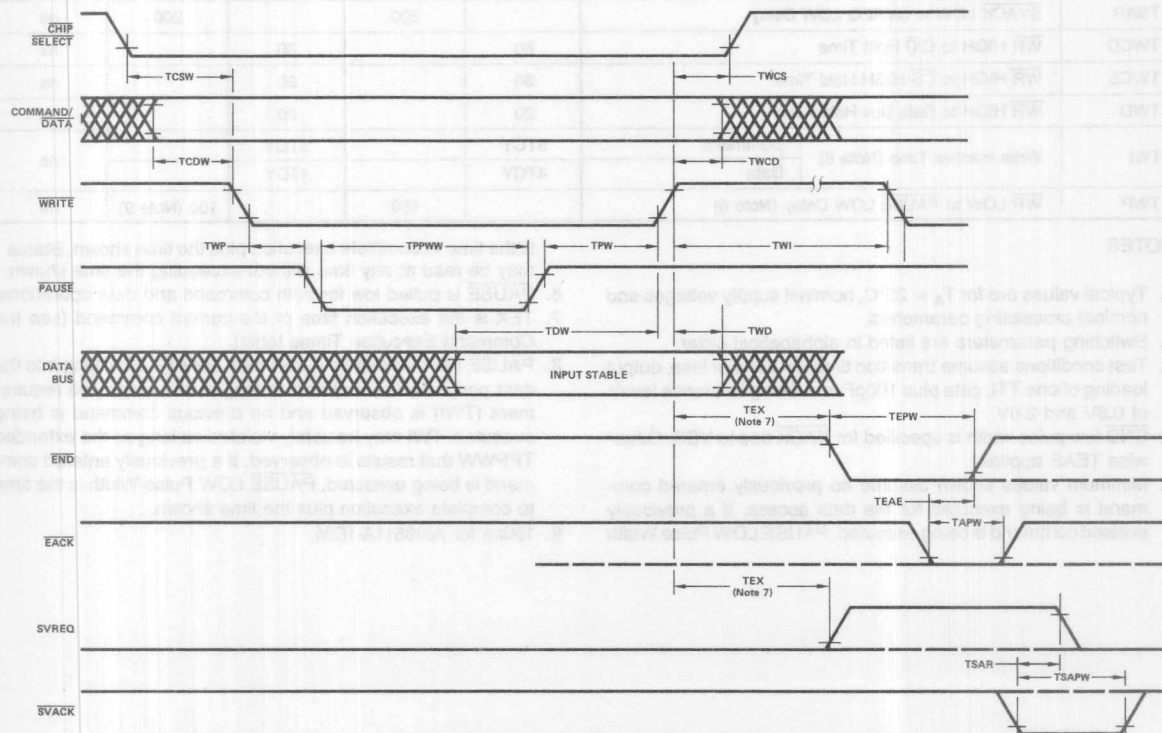
SWITCHING WAVEFORMS

READ OPERATIONS



MOS-048

WRITE OPERATIONS



MOS-049

APPLICATION INFORMATION

The diagram in Figure 2 shows the interface connections for the Am9511A APU with operand transfers handled by an Am9517 DMA controller, and CPU coordination handled by an Am9519 Interrupt Controller. The APU interrupts the CPU to indicate that a command has been completed. When the performance enhancements provided by the DMA and Interrupt

operations are not required, the APU interface can be simplified as shown in Figure 1. The Am9511A APU is designed with a general purpose 8-bit data bus and interface control so that it can be conveniently used with any general 8-bit processor.

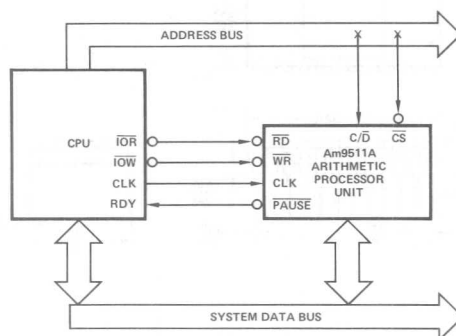


Figure 1. Am9511A Minimum Configuration Example.

MOS-050

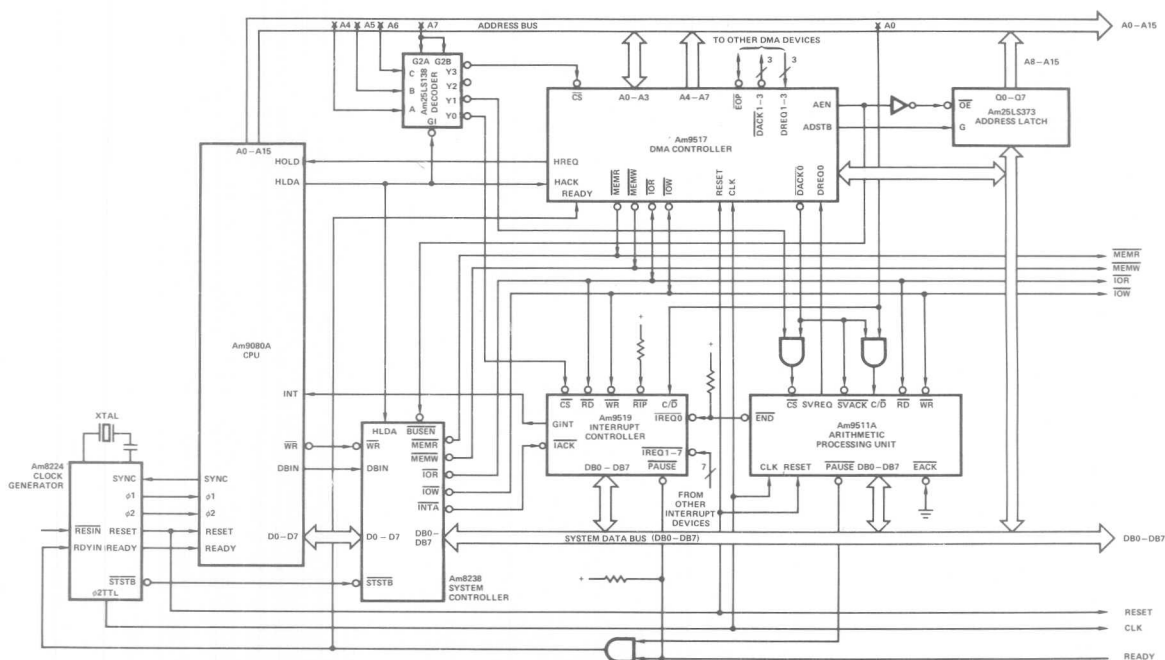


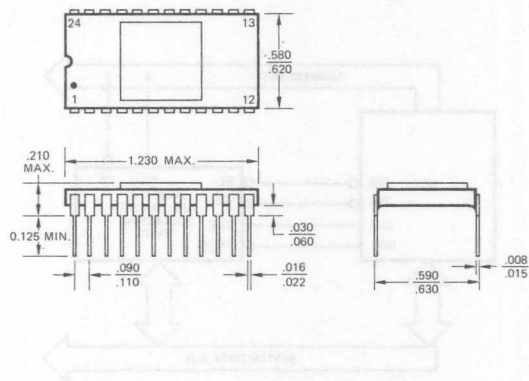
Figure 2. Am9511A High Performance Configuration Example.

MOS-051

PHYSICAL DIMENSIONS

Dual-In-Line

24-Pin Side-Brazed



Am9512

Floating-Point Processor

DISTINCTIVE CHARACTERISTICS

- Single (32-bit) and double (64-bit) precision capability
- Add, subtract, multiply and divide functions
- Compatible with proposed IEEE format
- Easy interfacing to microprocessors
- 8-bit data bus
- Standard 24-pin package
- 12V and 5V power supplies
- Stack oriented operand storage
- Direct memory access or programmed I/O Data Transfers
- End of execution signal
- Error interrupt
- All inputs and outputs TTL level compatible
- Advanced N-channel silicon gate MOS technology
- 100% MIL-STD-883 reliability assurance testing

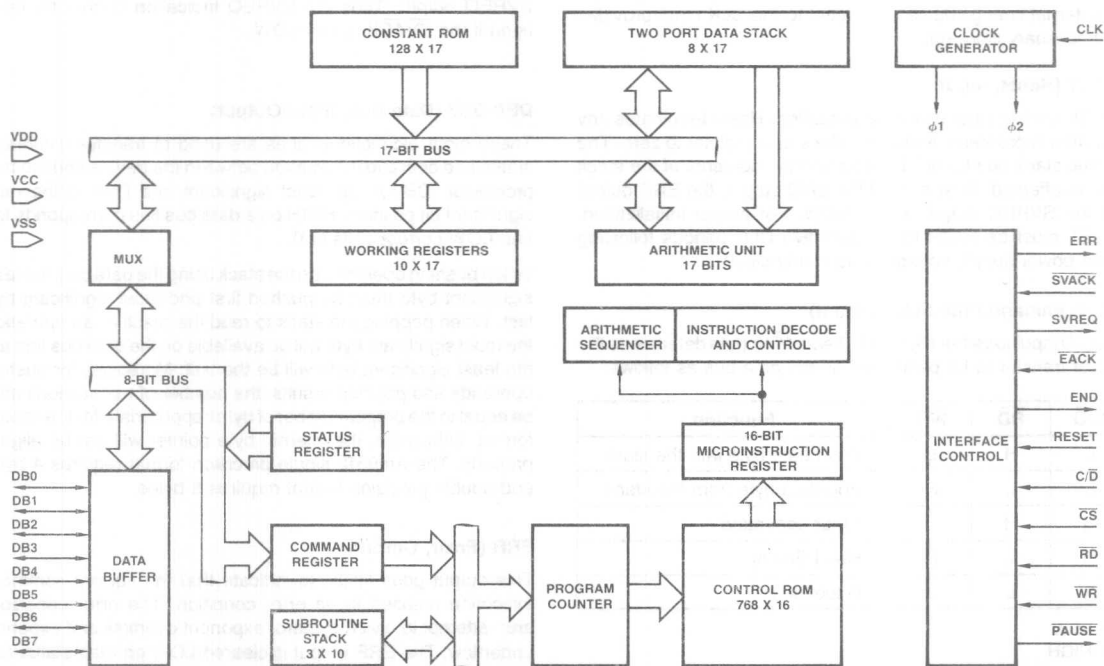
GENERAL DESCRIPTION

The Am9512 is a high performance floating-point processor unit (FPU). It provides single precision (32-bit) and double precision (64-bit) add, subtract, multiply and divide operations. It can be easily interfaced to enhance the computational capabilities of the host microprocessor.

The operand, result, status and command information transfers take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack by the host processor and a command is issued to perform an operation on the data stack. The results of this operation are available to the host processor by popping the stack.

Information transfers between the Am9512 and the host processor can be handled by using programmed I/O or direct memory access techniques. After completing an operation, the Am9512 activates an "end of execution" signal that can be used to interrupt the host processor.

BLOCK DIAGRAM

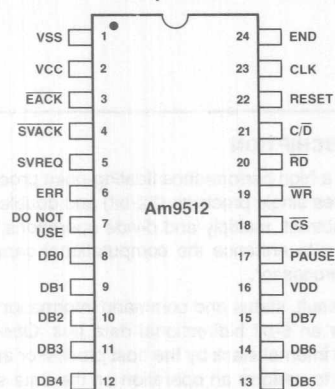


MOS-203

ORDERING INFORMATION

Package Type	Ambient Temperature	Maximum Clock Frequency	
		2MHz	3MHz
Hermetic DIP	$0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$	AM9512DC	AM9512-1DC
	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	AM9512DM	AM9512-1DM

CONNECTION DIAGRAM Top View



Note: Pin 1 is marked for orientation.

MOS-204

INTERFACE SIGNAL DESCRIPTION

VCC: +5V Power Supply

VDD: +12V Power Supply

VSS: Ground

CLK (Clock, Input)

An external timing source connected to the CLK input provides the necessary clocking.

RESET (Reset, Input)

A HIGH on this input causes initialization. Reset terminates any operation in progress, and clears the status register to zero. The internal stack pointer is initialized and the contents of the stack may be affected. After a reset the END output, the ERR output and the SVREQ output will be LOW. For proper initialization, RESET must be HIGH for at least five CLK periods following stable power supply voltages and stable clock.

C/D (Command/Data Select, Input)

The C/D input together with the RD and WR inputs determines the type of transfer to be performed on the data bus as follows:

C/D	RD	WR	Function
L	H	L	Push data byte into the stack
L	L	H	Pop data byte from the stack
H	H	L	Enter command
H	L	H	Read Status
X	L	L	Undefined

L = LOW

H = HIGH

X = DON'T CARE

END (End of Execution, Output)

A HIGH on this output indicates that execution of the current command is complete. This output will be cleared LOW by activating the EACK input LOW or performing any read or write operation or device initialization using the RESET. If EACK is tied LOW, the END output will be a pulse (see EACK description).

Reading the status register while a command execution is in progress is allowed. However any read or write operation clears

the flip-flop that generates the END output. Thus such continuous reading could conflict with internal logic setting of the END flip-flop at the end of command execution.

EACK (End Acknowledge, Input)

This input when LOW makes the END output go LOW. As mentioned earlier HIGH on the END output signals completion of a command execution. The END signal is derived from an internal flip-flop which is clocked at the completion of a command. This flip-flop is clocked to the reset state when EACK is LOW. Consequently, if EACK is tied LOW, the END output will be a pulse that is approximately one CLK period wide.

SVREQ (Service Request, Output)

A HIGH on this output indicates completion of a command. In this sense this output is the same as the END output. However, the SVREQ output will go HIGH at the completion of a command. This bit must be 1 for SVREQ to go HIGH. The SVREQ can be cleared (i.e., go LOW) by activating the SVACK input LOW or initializing the device using the RESET. Also, the SVREQ will be automatically cleared after completion of any command that has the service request bit as 0.

SVACK (Service Acknowledge, Input)

A LOW on this input clears SVREQ. If the SVACK input is permanently tied LOW, it will conflict with the internal setting of the SVREQ output. Thus the SVREQ indication cannot be relied upon if the SVACK is tied LOW.

DB0-DB7 (Data Bus, Input/Output)

These eight bidirectional lines are used to transfer command, status and operand information between the device and the host processor. DB0 is the least significant and DB7 is the most significant bit position. HIGH on a data bus line corresponds to 1 and LOW corresponds to 0.

When pushing operands on the stack using the data bus, the least significant byte must be pushed first and most significant byte last. When popping the stack to read the result of an operation, the most significant byte will be available on the data bus first and the least significant byte will be the last. Moreover, for pushing operands and popping results, the number of transactions must be equal to the proper number of bytes appropriate for the chosen format. Otherwise, the internal byte pointer will not be aligned properly. The Am9512 single precision format requires 4 bytes and double precision format requires 8 bytes.

ERR (Error, Output)

This output goes HIGH to indicate that the current command execution resulted in an error condition. The error conditions are: attempt to divide by zero, exponent overflow and exponent underflow. The ERR output is cleared LOW on read status register operation or upon RESET.

The ERR output is derived from the error bits in the status register. These error bits will be updated internally at an appropriate time during a command execution. Thus ERR output going HIGH may not correspond with the completion of a command. Reading of the status register can be performed while a command execution is in progress. However it should be noted that reading the status register clears the ERR output. Thus reading the status register while a command execution in progress may result in an internal conflict with the ERR output.

\overline{CS} (Chip Select, Input)

This input must be LOW to accomplish any read or write operation to the Am9512.

To perform a write operation, appropriate data is presented on DB0 through DB7 lines, appropriate logic level on the C/\overline{D} input and the \overline{CS} input is made LOW. Whenever \overline{WR} and \overline{RD} inputs are both HIGH and \overline{CS} is LOW, \overline{PAUSE} goes LOW. However actual writing into the Am9512 cannot start until \overline{WR} is made LOW. After initiating the write operation by the HIGH to LOW transition on the \overline{WR} input, the \overline{PAUSE} output will go HIGH indicating the write operation has been acknowledged. The \overline{WR} input can go HIGH after \overline{PAUSE} goes HIGH. The data lines, C/\overline{D} input and the \overline{CS} input can change when appropriate hold time requirements are satisfied. See write timing diagram for details.

To perform a read operation an appropriate logic level is established on the C/\overline{D} input and \overline{CS} is made LOW. The \overline{PAUSE} output goes LOW because \overline{WR} and \overline{RD} inputs are HIGH. The read operation does not start until the \overline{RD} input goes LOW. \overline{PAUSE} will go HIGH indicating that read operation is complete and the required information is available on the DB0 through DB7 lines. This information will remain on the data lines as long as \overline{RD} is LOW. The \overline{RD} input can return HIGH anytime after \overline{PAUSE} goes HIGH. The \overline{CS} input and C/\overline{D} input can change anytime after \overline{RD} returns HIGH. See read timing diagram for details. If the \overline{CS} is tied LOW permanently, \overline{PAUSE} will remain LOW until the next Am9512 read or write access.

 \overline{RD} (Read, Input)

A LOW on this input is used to read information from an internal location and gate that information onto the data bus. The \overline{CS} input must be LOW to accomplish the read operation. The C/\overline{D} input determines what internal location is of interest. See C/\overline{D} , \overline{CS} input descriptions and read timing diagram for details. If the \overline{END}

output was HIGH, performing any read operation will make the \overline{END} output go LOW after the HIGH to LOW transition of the \overline{RD} input (assuming \overline{CS} is LOW). If the \overline{ERR} output was HIGH performing a status register read operation will make the \overline{ERR} output LOW. This will happen after the HIGH to LOW transition of the \overline{RD} input (assuming \overline{CS} is LOW).

 \overline{WR} (Write, Input)

A LOW on this input is used to transfer information from the data bus into an internal location. The \overline{CS} must be LOW to accomplish the write operation. The C/\overline{D} determines which internal location is to be written. See C/\overline{D} , \overline{CS} input descriptions and write timing diagram for details.

If the \overline{END} output was HIGH, performing any write operation will make the \overline{END} output go LOW after the LOW to HIGH transition of the \overline{WR} input (assuming \overline{CS} is LOW).

 \overline{PAUSE} (Pause, Output)

This output is a handshake signal used while performing read or write transactions with the Am9512. If the \overline{WR} and \overline{RD} inputs are both HIGH, the \overline{PAUSE} output goes LOW with the \overline{CS} input in anticipation of a transaction. If \overline{WR} goes LOW to initiate a write transaction with proper signals established on the DB0-DB7, C/\overline{D} inputs, the \overline{PAUSE} will return HIGH indicating that the write operation has been accomplished. The \overline{WR} can be made HIGH after this event. On the other hand, if a read operation is desired, the \overline{RD} input is made LOW after activating \overline{CS} LOW and establishing proper C/\overline{D} input. (The \overline{PAUSE} will go LOW in response to \overline{CS} going LOW.) The \overline{PAUSE} will return HIGH indicating completion of read. The \overline{RD} can return HIGH after this event. It should be noted that a read or write operation can be initiated without any regard to whether a command execution is in progress or not. Proper device operation is assured by obeying the \overline{PAUSE} output indication as described.

FUNCTIONAL DESCRIPTION

Major functional units of the Am9512 are shown in the block diagram. The Am9512 employs a microprogram controlled stack oriented architecture with 17-bit wide data paths.

The Arithmetic Unit receives one of its operands from the Operand Stack. This stack is an eight word by 17-bit two port memory with last in – first out (LIFO) attributes. The second operand to the Arithmetic Unit is supplied by the internal 17-bit bus. In addition to supplying the second operand, this bidirectional bus also carries the results from the output of the Arithmetic Unit when required. Writing into the Operand Stack takes place from this internal 17-bit bus when required. Also connected to this bus are the Constant ROM and Working Registers. The ROM provides the required constants to perform the mathematical operations while the Working Registers provide storage for the intermediate values during command execution.

Communication between the external world and the Am9512 takes place on eight bidirectional input/output lines, DB0 through

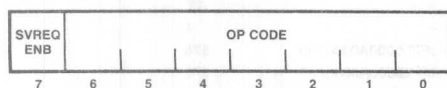
DB7 (Data Bus). These signals are gated to the internal 8-bit bus through appropriate interface and buffer circuitry. Multiplexing facilities exist for bidirectional communication between the internal eight and 17-bit buses. The Status Register and Command Register are also located on the 8-bit bus.

The Am9512 operations are controlled by the microprogram contained in the Control ROM. The Program Counter supplies the microprogram addresses and can be partially loaded from the Command Register. Associated with the Program Counter is the Subroutine Stack where return addresses are held during subroutine calls in the microprogram. The Microinstruction Register holds the current microinstruction being executed. The register facilitates pipelined microprogram execution. The Instruction Decode logic generates various internal control signals needed for the Am9512 operation.

The Interface Control logic receives several external inputs and provides handshake related outputs to facilitate interfacing the Am9512 to microprocessors.

COMMAND FORMAT

The Operation of the Am9512 is controlled from the host processor by issuing instructions called commands. The command format is shown below:



The command consists of 8 bits; the least significant 7 bits specify the operation to be performed as detailed in the accompanying

table. The most significant bit is the Service Request Enable bit. This bit must be a 1 if SVREQ is to go high at end of executing a command.

The Am9512 commands fall into three categories: Single precision arithmetic, double precision arithmetic and data manipulation. There are four arithmetic operations that can be performed with single precision (32-bit), or double precision (64-bit) floating-point numbers: add, subtract, multiply and divide. These operations require two operands. The Am9512 assumes that these operands are located in the internal stack as Top of Stack

(TOS) and Next on Stack (NOS). The result will always be returned to the previous NOS which becomes the new TOS. Results from an operation are of the same precision and format as the operands. The results will be rounded to preserve the accuracy. The actual data formats and rounding procedures are described in a later section. In addition to the arithmetic operations, the Am9512 implements eight data manipulating operations. These include changing the sign of a double or single precision

operand located in TOS, exchanging single precision operands located at TOS and NOS, as well as copying and popping single or double precision operands. See also the sections on status register and operand formats.

The Execution times of the Am9512 commands are all data dependent. Table 2 shows one example of each command execution time:

Table 1. Command Decoding Table.

Command Bits								Mnemonic	Description
7	6	5	4	3	2	1	0		
X	0	0	0	0	0	0	1	SADD	Add TOS to NOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	0	1	0	SSUB	Subtract TOS from NOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	0	1	1	SMUL	Multiply NOS by TOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	1	0	0	SDIV	Divide NOS by TOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	1	0	1	CHSS	Change sign of TOS Single Precision operand.
X	0	0	0	0	1	1	0	PTOS	Push Single Precision operand on TOS to NOS.
X	0	0	0	0	1	1	1	POPS	Pop Single Precision operand from TOS. NOS becomes TOS.
X	0	0	0	1	0	0	0	XCHS	Exchange TOS with NOS Single Precision.
X	0	1	0	1	1	0	1	CHSD	Change sign of TOS Double Precision operand.
X	0	1	0	1	1	1	0	PTOD	Push Double Precision operand on TOS to NOS.
X	0	1	0	1	1	1	1	POPD	Pop Double Precision operand from TOS. NOS becomes TOS.
X	0	0	0	0	0	0	0	CLR	CLR status.
X	0	1	0	1	0	0	1	DADD	Add TOS to NOS Double Precision and result to NOS. Pop stack.
X	0	1	0	1	0	1	0	DSUB	Subtract TOS from NOS Double Precision and result to NOS. Pop stack.
X	0	1	0	1	0	1	1	DMUL	Multiply NOS by TOS Double Precision and result to NOS. Pop stack.
X	0	1	0	1	1	0	0	DDIV	Divide NOS by TOS Double Precision and result to NOS. Pop Stack.

Notes: X = Don't Care

Operation for bit combinations not listed above is undefined.

Table 2. Am9512 Execution Time in Cycles.

Single Precision				Double Precision			
	Min	Typ	Max		Min	Typ	Max
Add	58	220	512	Add	578	1200	3100
Subtract	56	220	512	Subtract	578	1200	3100
Multiply	192	220	254	Multiply	1720	1770	1860
Divide	228	240	264	Divide	4560	4920	5120

Note: Typical for add and subtract, assumes the operands are within six decimal orders of magnitude. Max is derived from the maximum execution time of 1000 executions with random 32-bit or 64-bit patterns.

Table 3. Some Execution Examples.

Command	TOS	NOS	Result	Clock periods
SADD	3F800000	3F800000	40000000	58
SSUB	3F800000	3F800000	00000000	56
SMUL	40400000	3FC00000	40900000	198
SDIV	40000000	3F800000	3F000000	228
CHSS	3F800000	—	BF800000	10
PTOS	3F800000	—	—	16
POPS	3F800000	—	—	14
XCHS	3F800000	40000000	—	26
CHSD	3FF0000000000000	—	BFF0000000000000	24
PTOD	3FF0000000000000	—	—	40
POPD	3FF0000000000000	—	—	26
CLR	3FF0000000000000	—	—	4
DADD	3FF00000A0000000	8000000000000000	3FF00000A0000000	578
DSUB	3FF00000A0000000	8000000000000000	3FF00000A0000000	578
DMUL	BFF8000000000000	3FF8000000000000	C002000000000000	1748
DDIV	BFF8000000000000	3FF8000000000000	BFF0000000000000	4560

Note: TOS, NOS and Result are in hexadecimal; Clock period is in decimal.

COMMAND INITIATION

After properly positioning the required operands in the stack, a command may be issued. The procedure for initiating a command execution is as follows:

1. Establish appropriate command on the DB0-DB7 lines.
2. Establish HIGH on the C/\bar{D} input.
3. Establish LOW on the \overline{CS} input. Whenever \overline{WR} and \overline{RD} inputs are HIGH the \overline{PAUSE} output follows the \overline{CS} input. Hence \overline{PAUSE} will become LOW.
4. Establish LOW on the \overline{WR} input after an appropriate set up time (see timing diagrams).
5. Sometime after the HIGH to LOW level transition of \overline{WR} input, the \overline{PAUSE} output will become HIGH to acknowledge the write operation. The \overline{WR} input can return to HIGH anytime after \overline{PAUSE} goes HIGH. The DB0-DB7, C/\bar{D} and \overline{CS} inputs are allowed to change after the hold time requirements are satisfied (see timing diagram).

An attempt to issue a new command while the current command execution is in progress is allowed. Under these circumstances, the \overline{PAUSE} output will not go HIGH until the current command execution is completed.

OPERAND ENTRY

The Am9512 commands operate on the operands located at the TOS and NOS and results are returned to the stack at NOS and then popped to TOS. The operands required for the Am9512 are one of two formats – single precision floating-point (4 bytes) or double precision floating-point (8 bytes). The result of an operation has the same format as the operands. In other words, operations using single precision quantities always result in a single precision result while operations involving double precision quantities will result in double precision result.

Operands are always entered into the stack least significant byte first and most significant byte last. The following procedure must be followed to enter operands into the stack:

1. The lower significant operand byte is established on the DB0-DB7 lines.
2. A LOW is established on the C/\bar{D} input to specify that data is to be entered into the stack.
3. The \overline{CS} input is made LOW. Whenever the \overline{WR} and \overline{RD} inputs are HIGH, the \overline{PAUSE} output will follow the \overline{CS} input. Thus \overline{PAUSE} output will become LOW.
4. After appropriate set up time (see timing diagrams), the \overline{WR} input is made LOW.
5. Sometime after this event, \overline{PAUSE} will return HIGH to indicate that the write operation has been acknowledged.
6. Anytime after the \overline{PAUSE} output goes HIGH the \overline{WR} input can be made HIGH. The DB0-DB7, C/\bar{D} and \overline{CS} inputs can change after appropriate hold time requirements are satisfied (see timing diagrams).

The above procedure must be repeated until all bytes of the operand are pushed into the stack. It should be noted that for single precision operands 4 bytes should be pushed and 8 bytes must be pushed for double precision. Not pushing all the bytes of a quantity will result in byte pointer misalignment.

The Am9512 stack can accommodate 4 single precision quantities or 2 double precision quantities. Pushing more quantities than the capacity of the stack will result in loss of data which is usual with any LIFO stack.

REMOVING THE RESULTS

Result from an operation will be available at the TOS. Results can be transferred from the stack to the data bus by reading the stack.

When the stack is popped for results, the most significant byte is available first and the least significant byte last. A result is always of the same precision as the operands that produced it. Thus when the result is taken from the stack, the total number of bytes popped out should be appropriate with the precision – single precision results are 4 bytes and double precision results are 8 bytes. The following procedure must be used for reading the result from the stack:

1. A LOW is established on the C/\bar{D} input.
2. The \overline{CS} input is made LOW. When \overline{WR} and \overline{RD} inputs are both HIGH, the \overline{PAUSE} output follows the \overline{CS} input, thus \overline{PAUSE} will be LOW.
3. After appropriate set up time (see timing diagrams), the \overline{RD} input is made LOW.
4. Sometime after this, \overline{PAUSE} will return HIGH indicating that the data is available on the DB0-DB7 lines. This data will remain on the DB0-DB7 lines as long as the \overline{RD} input remains LOW.
5. Anytime after \overline{PAUSE} goes HIGH, the \overline{RD} input can return HIGH to complete transaction.
6. The \overline{CS} and C/\bar{D} inputs can change after appropriate hold time requirements are satisfied (see timing diagram).
7. Repeat this procedure until all bytes appropriate for the precision of the result are popped out.

Reading of the stack does not alter its data; it only adjusts the byte pointer. If more data is popped than the capacity of the stack, the internal byte pointer will wrap around and older data will be read again, consistent with the LIFO stack.

READING STATUS REGISTER

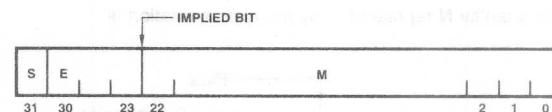
The Am9512 status register can be read without any regard to whether a command is in progress or not. The only implication that has to be considered is the effect this might have on the END and ERR outputs discussed in the signal descriptions.

The following procedure must be followed to accomplish status register reading.

1. Establish HIGH on the C/\bar{D} input.
2. Establish LOW on the \overline{CS} input. Whenever \overline{WR} and \overline{RD} inputs are HIGH, \overline{PAUSE} will follow the \overline{CS} input. Thus, \overline{PAUSE} will go LOW.
3. After appropriate set up time (see timing diagram) \overline{RD} is made LOW.
4. Sometime after the HIGH to LOW transition of \overline{RD} , \overline{PAUSE} will become HIGH indicating that status register contents are available on the DB0-DB7 lines. These lines will contain this information as long as \overline{RD} is LOW.
5. The \overline{RD} input can be returned HIGH anytime after \overline{PAUSE} goes HIGH.
6. The C/\bar{D} input and \overline{CS} input can change after satisfying appropriate hold time requirements (see timing diagram).

DATA FORMATS

The Am9512 handles floating-point quantities in two different formats – single precision and double precision. The single precision quantities are 32-bits long as shown below.



Bit 31:

S = Sign of the mantissa. 1 represents negative and 0 represents positive.

Bits 23-30

E = These 8-bits represent a biased exponent. The bias is $2^7 - 1 = 127$

Bits 0-22

M = 23-bit mantissa. Together with the sign bit, the mantissa represents a signed fraction in sign-magnitude notation. There is an implied 1 beyond the most significant bit (bit 22) of the mantissa. In other words, the mantissa is assumed to be a 24-bit normalized quantity and the most significant bit which will always be 1 due to normalization is implied. The Am9512 restores this implied bit internally before performing arithmetic; normalizes the result and strips the implied bit before returning the results to the external data bus. The binary point is between the implied bit and bit 22 of the mantissa.

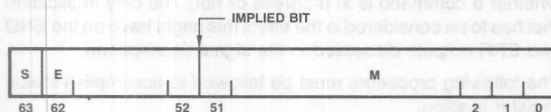
The quantity N represented by the above notation is

$$N = (-1)^S \cdot 2^{E - (2^7 - 1)} \cdot (1.M)$$

Bias
Binary Point

Provided $E \neq 0$ or all 1's.

A double precision quantity consists of the mantissa sign bit(s), an 11 bit biased exponent (E), and a 52-bit mantissa (M). The bias for double precision quantities is $2^{10} - 1$. The double precision format is illustrated below.

**Bit 63:**

S = Sign of the mantissa. 1 represents negative and 0 represents positive.

Bits 52-62

E = These 11 bits represent a biased exponent. The bias is $2^{10} - 1 = 1023$.

Bit 0-51

M = 52-bit mantissa. Together with the sign bit, the mantissa represents a signed fraction in sign-magnitude notation. There is an implied 1 beyond the most significant bit (bit 51) of the mantissa. In other words, the mantissa is assumed to be a 53-bit normalized quantity and the most significant bit, which will always be 1 due to normalization, is implied. The Am9512 restores this implied bit internally before performing arithmetic; normalizes the result and strips the implied bit before returning the result to the external data bus. The binary point is between the implied bit and bit 51 of the mantissa.

The quantity N represented by the above notation is

$$N = (-1)^S \cdot 2^{E - (2^{10} - 1)} \cdot (1.M)$$

Bias
Binary point

Provided $E \neq 0$ or all 1's.

STATUS REGISTER

The Am9512 contains an 8-bit status register with the following format.

BUSY	SIGN S	ZERO Z	RESERVED	DIVIDE EXCEPTION D	EXPONENT UNDERFLOW U	EXPONENT OVERFLOW V	RESERVED
7	6	5	4	3	2	1	0

Bit 0 and bit 4 are reserved. Occurrence of exponent overflow (V), exponent underflow (U) and divide exception (D) are indicated by bits 1, 2 and 3 respectively. An attempt to divide by zero is the only divide exception. Bits 5 and 6 represent a zero result and the sign of a result respectively. Bit 7 (Busy) of the status register indicates if the Am9512 is currently busy executing a command. All the bits are initialized to zero upon reset. Also, executing a CLR (Clear Status) command will result in all zero status register bits. A zero in Bit 7 indicates that the Am9512 is not busy and a new command may be initiated. As soon as a new command is issued, Bit 7 becomes 1 to indicate the device is busy and remains 1 until the command execution is complete, at which time it will become 0. As soon as a new command is issued, status register bits 0, 1, 2, 3, 4, 5 and 6 are cleared to zero. The status bits will be set as required during the command execution. Hence, as long as bit 7 is 1, the remainder of the status register bit indications should not be relied upon unless the ERR occurs. The following is a detailed status bit description.

Bit 0 Reserved

Bit 1 Exponent overflow (V): When 1, this bit indicates that exponent overflow has occurred. Cleared to zero otherwise.

Bit 2 Exponent Underflow (U): When 1, this bit indicates that exponent underflow has occurred. Cleared to zero otherwise.

Bit 3 Divide Exception (D): When 1, this bit indicates that an attempt to divide by zero is made. Cleared to zero otherwise.

Bit 4 Reserved

Bit 5 Zero (Z): When 1, this bit indicates that the result returned to TOS after a command is all zeros. Cleared to zero otherwise.

Bit 6 Sign (S): When 1, this bit indicates that the result returned to TOS is negative. Cleared to zero otherwise.

Bit 7 Busy: When 1, this bit indicates the Am9512 is in the process of executing a command. It will become zero after the command execution is complete.

All other status register bits are valid when the Busy bit is zero.

ALGORITHMS OF FLOATING-POINT ARITHMETIC**1. Floating Point to Decimal Conversion**

As an introduction to floating-point arithmetic, a brief description of the Decimal equivalent of the Am9512 floating-point format should help the reader to understand and verify the validity of the arithmetic operations. The Am9512 single precision format is used for the following discussions. With a minor modification of the field lengths, the discussion would also apply to the double precision format.

There are three parts in a floating point number:

- The sign – the sign applies to the sign of the number. Zero means the number is positive or zero. One means the number is negative.

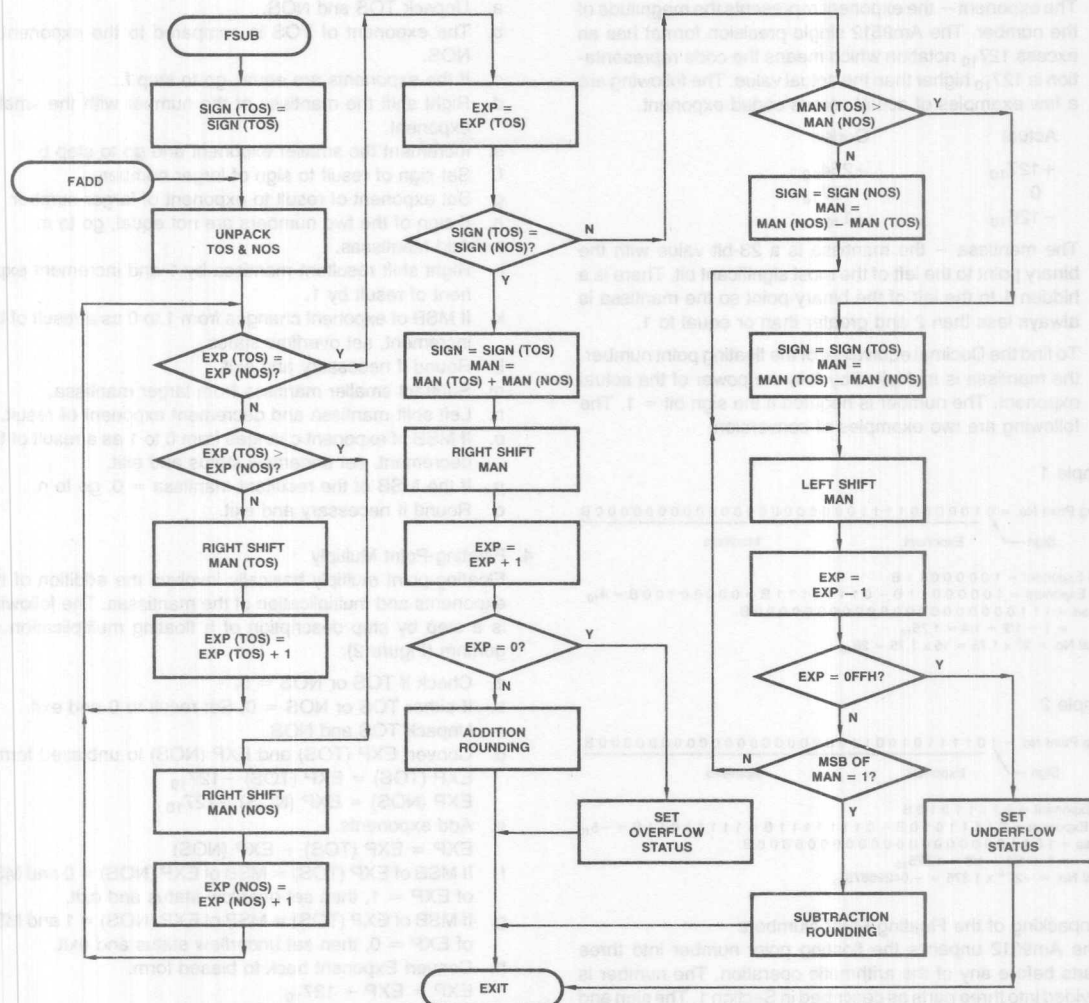


Figure 1. Conceptual Floating-Point Addition/Subtraction.

MOS-205

- h. Add bias to exponent of result.
 $EXP = EXP + 127_{10}$
- i. If sign of TOS = sign of NOS, set sign of result to 0, else set sign of result to 1.
- j. Divide mantissa of NOS by mantissa of TOS.
- k. If MSB = 0, left shift mantissa and decrement exponent of resultant, else go to n.
- l. If MSB of exponent changes from 0 to 1 as a result of the decrement, set underflow status.
- m. Go to k.
- n. Round if necessary and exit.

The algorithms described above provide the user a means of verifying the validity of the result. They do not necessarily reflect the exact internal sequence of the Am9512.

6. Rounding

The Am9512 adopts a rounding algorithm that is consistent with the Intel® standard for floating-point arithmetic. The following description is an excerpt from the paper published in proceedings of Compsac 77, November 1977, pp. 107-112 by Dr. John F. Palmer of Intel Corporation.

The method used for doing the rounding during floating-point arithmetic is known as "Round to Even", i.e., if the resultant number is exactly halfway between two floating point numbers, the number is rounded to the nearest floating-point number whose LSB of the mantissa is 0. In order to simplify the explanation, the algorithms will be illustrated with 4-bit arithmetic. The existence of an accumulator will be assumed as shown:

OF	B1	B2	B3	B4	G	R	ST
----	----	----	----	----	---	---	----

The bit labels denote:

- OF – The overflow bit
- B1-B4 – The 4 mantissa bits
- G – The Guard bit
- R – The Rounding bit
- ST – The "Sticky" bit

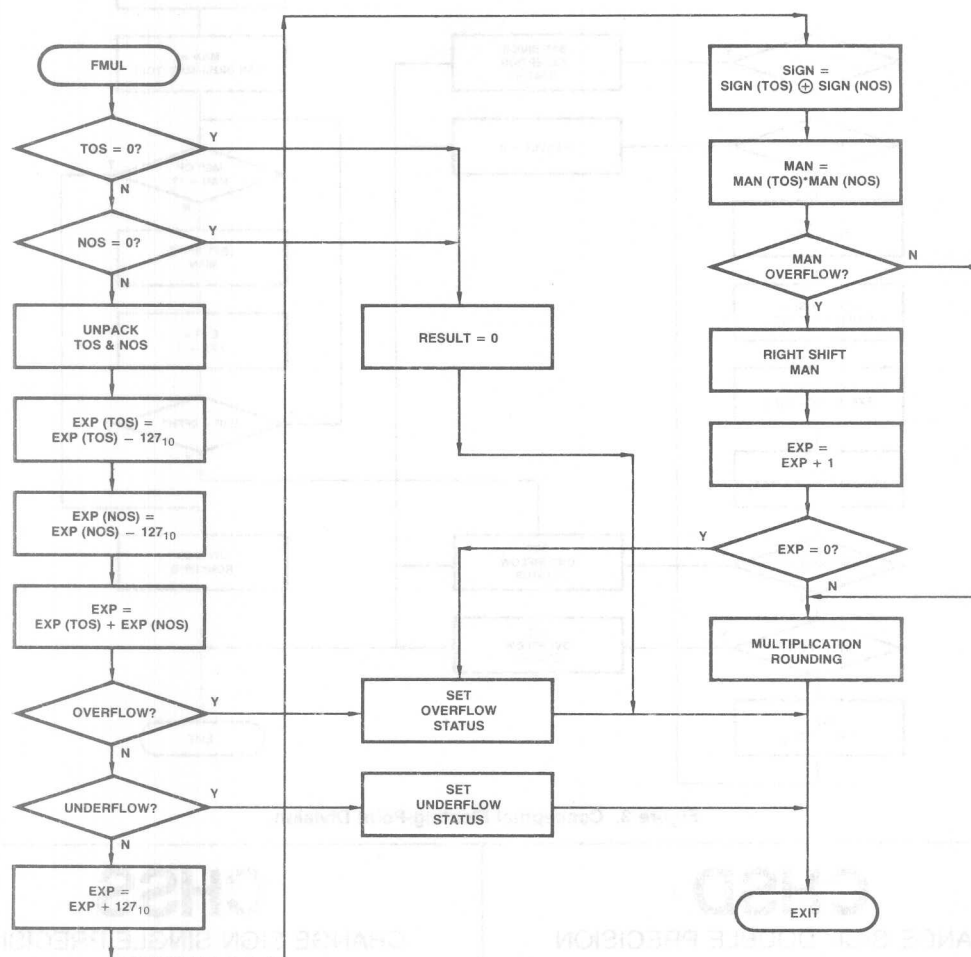


Figure 2. Conceptual Floating-Point Multiplication.

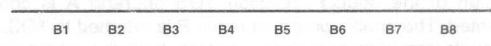
MOS-206

The Sticky bit is set to one if any ones are shifted right of the rounding bit in the process of denormalization. If the Sticky bit becomes set, it remains set throughout the operation. All shifting in the Accumulator involves the OF, G, R and ST bits. The ST bit is not affected by left shifts but, zeros are introduced into OF by right shifts.

Rounding during addition of magnitudes – add 1 to the G position, then if G=R=ST=0, set B4 to 0 (“Rounding to Even”).

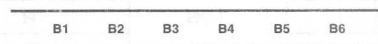
Rounding during subtraction of magnitudes – if more than one left shift was performed, no rounding is needed, otherwise round the same way as addition of magnitudes.

Rounding during multiplication – let the normalized double length product be:



Then G=B5, R=B6, ST=B7 V B8. The rounding is then performed as in addition of magnitudes.

Rounding during division – let the first six bits of the normalized quotient be



Then G=B5, R=B6, ST=0 if and only if remainder = 0. The rounding is then performed as in addition of magnitudes.

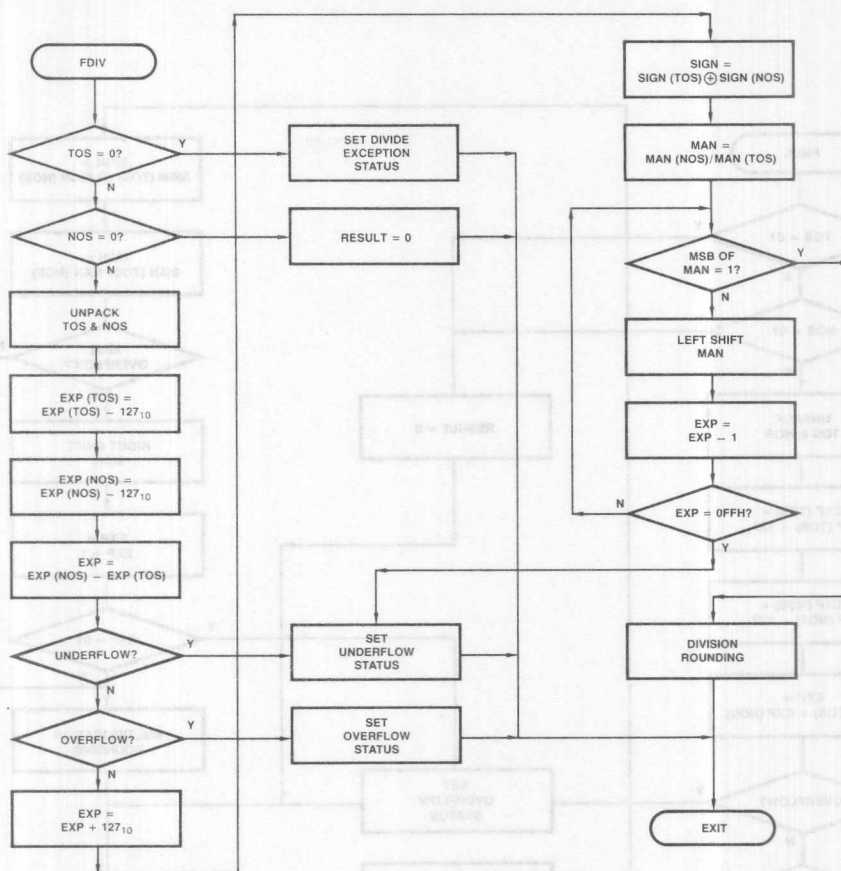


Figure 3. Conceptual Floating-Point Division.

MOS-207

CHSD

CHANGE SIGN DOUBLE PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	1	0	1	1	0	1

Hex Coding: AD IF SRE = 1
2D IF SRE = 0

Execution Time: See Table 2

Description:

The sign of the double precision TOS operand A is complemented. The double precision result R is returned to TOS. If the double precision operand A is zero, then the sign is not affected. The status bit S and Z indicate the sign of the result and if the result is zero. The status bits U, V and D are always cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS

BEFORE		AFTER	
A	TOS	R	
B	NOS	B	

CHSS

CHANGE SIGN SINGLE PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	0	0	0	1	0	1

Hex Coding: 85 IF SRE = 1
05 IF SRE = 0

Execution Time: See Table 2

Description:

The sign of the single precision operand A at TOS is complemented. The single precision result R is returned to TOS. If the exponent field of A is zero, all bits of R will be zeros. The status bits S and Z indicate the sign of the result and if the result is zero. The status bits U, V and D are cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS

BEFORE		AFTER	
A	← TOS →	R	
B	← NOS →	B	
C		C	
D		D	

CLR

CLEAR STATUS

7 6 5 4 3 2 1 0

Binary Coding: SRE 0 0 0 0 0 0 0 0

Hex Coding: 80 IF SRE = 1

00 IF SRE = 0

Execution Time: 4 clock cycles

Description:

The status bits S, Z, D, U, V are cleared to zero. The stack is not affected. This essentially is a no operation command as far as operands are concerned.

Status Affected: S, Z, D, U, V always zero.

DSUB

DOUBLE PRECISION FLOATING-POINT SUBTRACT

7 6 5 4 3 2 1 0

Binary Coding: SRE 0 1 0 1 0 1 0 0

Hex Coding: AA IF SRE = 1

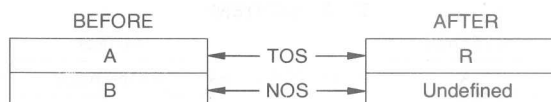
2A IF SRE = 0

Execution Time: See Table 2

Description:

The double precision operand A at TOS is subtracted from the double precision operand B at NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS

DADD

DOUBLE PRECISION FLOATING-POINT ADD

7 6 5 4 3 2 1 0

Binary Coding: SRE 0 1 0 1 0 0 0 1

Hex Coding: A9 IF SRE = 1

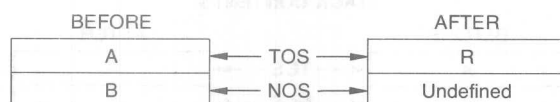
29 IF SRE = 0

Execution Time: See Table 2

Description:

The double precision operand A from TOS is added to the double precision operand B from NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS

DMUL

DOUBLE PRECISION FLOATING-POINT MULTIPLY

7 6 5 4 3 2 1 0

Binary Coding: SRE 0 1 0 1 0 1 1 1

Hex Coding: AB IF SRE = 1

2B IF SRE = 0

Execution Time: See Table 2

Description:

The double precision operand A from TOS is multiplied by the double precision operand B from NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS

DDIV

DOUBLE PRECISION FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Code:	SRE	0	1	0	1	1	0	0

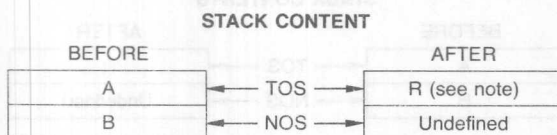
Hex Coding: AC IF SRE = 1
2C IF SRE = 0

Execution Time: See Table 2

Description:

The double precision operand B from NOS is divided by the double precision operand A from TOS. The result (quotient) is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, D, U and V are affected to report sign of the result, if the result is zero, attempt to divide by zero, exponent underflow and exponent overflow respectively.

Status Affected: S, Z, D, U, V



Note: If A is zero, then R = B (Divide exception).

SADD

SINGLE PRECISION FLOATING-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	0	1

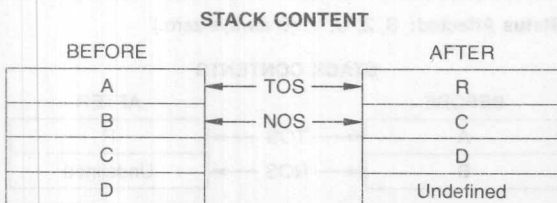
Hex Coding: 81 IF SRE = 1
01 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand A from TOS is added to the single precision operand B from NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)



SSUB

SINGLE PRECISION FLOATING-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	1	0

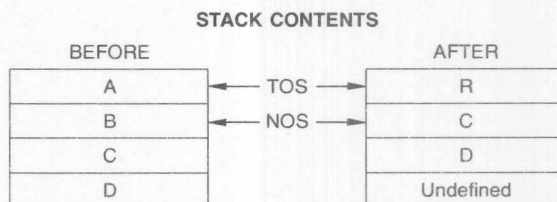
Hex Coding: 82 IF SRE = 1
02 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand A at TOS is subtracted from the single precision operand B at NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)



SMUL

SINGLE PRECISION FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	1	1

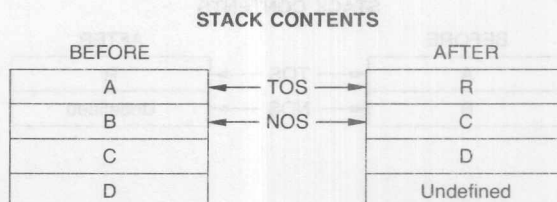
Hex Coding: 83 IF SRE = 1
03 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand A from TOS is multiplied by the single precision operand B from NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)



SDIV

SINGLE PRECISION FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	0	0

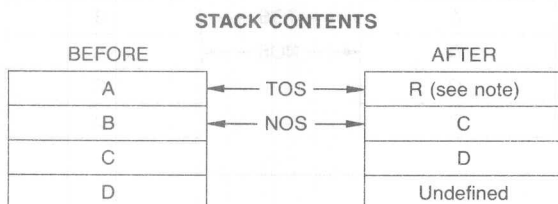
Hex Coding: 84 IF SRE = 1
04 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand B from NOS is divided by the single precision operand A from TOS. The result (quotient) is rounded to obtain the final result R which is returned to TOS. The status bits S, Z, D, U and V are affected to report the sign of the result, if the result is zero, attempt to divide by zero, exponent underflow and exponent overflow respectively.

Status Affected: S, Z, D, U, V



Note: If exponent field of A is zero then R = B (Divide exception).

POPS

POP STACK SINGLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	1	1

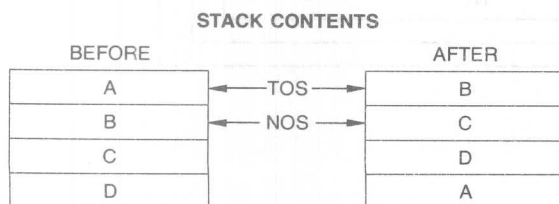
Hex Coding: 87 IF SRE = 1
07 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand A is popped from the stack. The internal stack control mechanism is such that A will be written at the bottom of the stack. The status bits S and Z are affected to report the sign of the new operand at TOS and if it is zero, respectively. The status bits U, V and D will be cleared to zero. Note that only the exponent field of the new TOS is checked for zero, if it is zero status bit Z will set to 1.

Status Affected: S, Z, (U, V, D always zero.)



PTOD

PUSH STACK DOUBLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	1	1	0

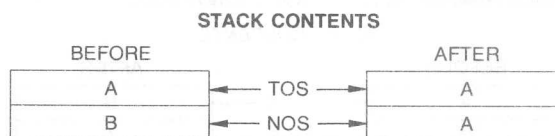
Hex Coding: AE IF SRE = 1
2E IF SRE = 0

Execution Time: See Table 2

Description:

The double precision operand A from the TOS is pushed back on to the stack. This is effectively a duplication of A into two consecutive stack locations. The status S and Z are affected to report sign of the new TOS and if the new TOS is zero respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z, (U, V, D always zero.)



PTOS

PUSH STACK SINGLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	1	0

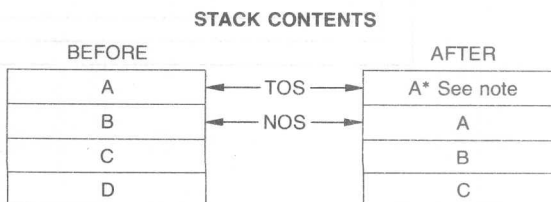
Hex Coding: 86 IF SRE = 1
06 IF SRE = 0

Execution Time: See Table 2

Description:

This instruction effectively pushes the single precision operand from TOS on to the stack. This amounts to duplicating the operand at two locations in the stack. However, if the operand at TOS prior to the PTOS command has only its exponent field as zero, the new content of the TOS will all be zeroes. The contents of NOS will be an exact copy of the old TOS. The status bits S and Z are affected to report the sign of the new TOS and if the content of TOS is zero, respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z, (U, V, D always zero.)



Note: A* = A if Exponent field of A is not zero.
A* = 0 if Exponent field of A is zero.

POPD

POP STACK DOUBLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	1	1	1

Hex Coding: AF IF SRE = 1
2F IF SRE = 0

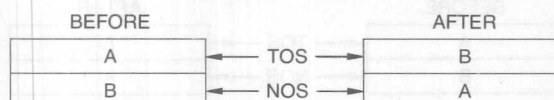
Execution Time: See Table 2

Description:

The double precision operand A is popped from the stack. The internal stack control mechanism is such that A will be written at the bottom of the stack. This operation has the same effect as exchanging TOS and NOS. The status bits S and Z are affected to report the sign of the new operand at TOS and if it is zero, respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z (U, V and D always zero.)

STACK CONTENTS



XCHS

EXCHANGE TOS AND NOS SINGLE-PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	1	0	0	0

Hex Coding: 88 IF SRE = 1
08 IF SRE = 0

Execution Time: See Table 2

Description:

The single precision operand A at the TOS and the single precision operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All other operands are unchanged.

Status Affected: S, Z (U, V and D always zero.)

STACK CONTENTS

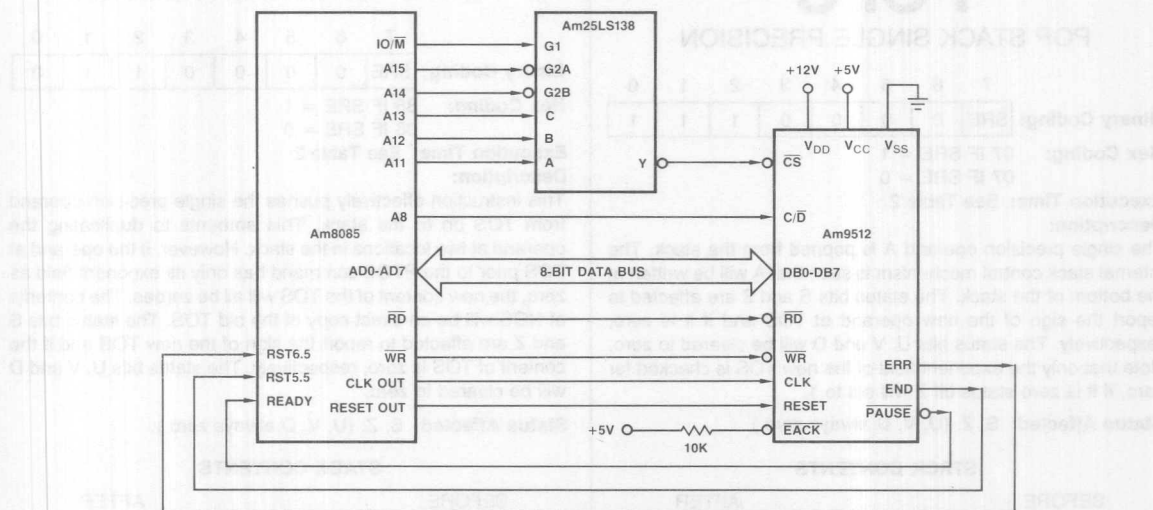
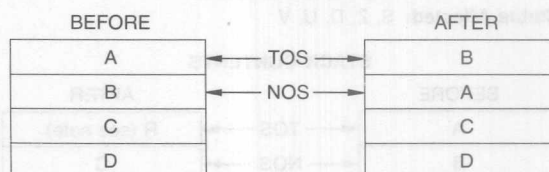


Figure 1. Am9512 to Am8085 Interface.

MOS-213

MAXIMUM RATINGS beyond which useful life may be impaired

Storage Temperature	−65°C to +150°C
Ambient Temperature Under Bias	−55°C to +125°C
VDD with Respect to VSS	−0.5V to +15.0V
VCC with Respect to VSS	−0.5V to +7.0V
All Signal Voltages with Respect to VSS	−0.5V to +7.0V
Power Dissipation (Package Limitation)	2.0W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VSS	VCC	VDD
Am9512DC	0°C ≤ T _A ≤ +70°C	0V	+5.0V ±5%	+12V ±5%
Am9512DM	−55°C ≤ T _A ≤ +125°C	0V	+5.0V ±10%	+12V ±10%

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
VOH	Output HIGH Voltage	IOH = −200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T _A = +25°C		50	90	mA
		T _A = 0°C			95	
		T _A = −55°C			100	
IDD	VDD Supply Current	T _A = +25°C		50	90	mA
		T _A = 0°C			95	
		T _A = −55°C			100	
CO	Output Capacitance			8	10	pF
CI	Input Capacitance	fc = 1.0MHz, Inputs = 0V		5	8	pF
CIO	I/O Capacitance			10	12	pF

Am9512

SWITCHING CHARACTERISTICS

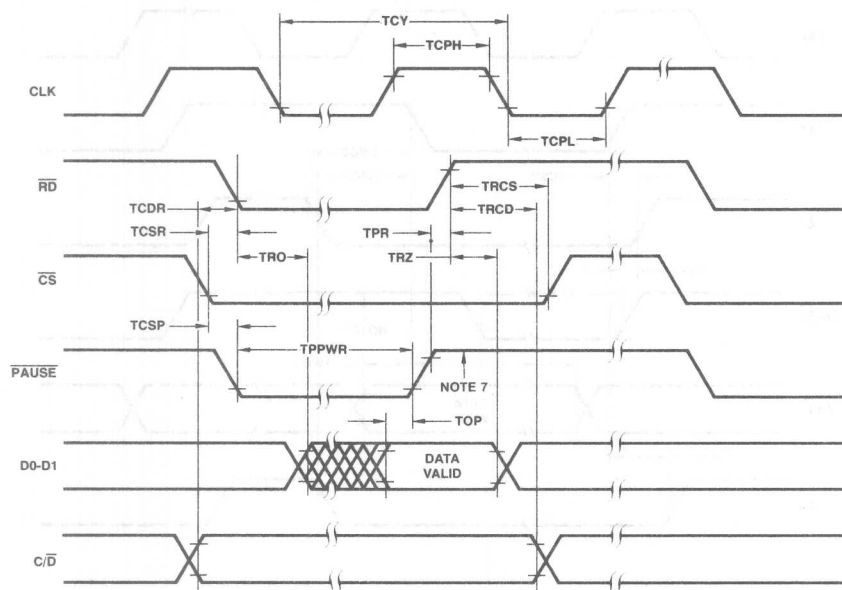
Parameters		Description	Am9512DC		Am9512-1DC		Units
			Min	Max	Min	Max	
TAPW	$\overline{\text{EACK}}$ LOW Pulse Width		100		75		ns
TCDR	C/ $\overline{\text{D}}$ to $\overline{\text{RD}}$ LOW Set-up Time		0		0		ns
TCDW	C/ $\overline{\text{D}}$ to $\overline{\text{WR}}$ LOW Set-up Time		0		0		ns
TCPH	Clock Pulse HIGH Width		200	500	140	500	ns
TCPL	Clock Pulse LOW Width		240		160		ns
TCSP	$\overline{\text{CS}}$ LOW to $\overline{\text{PAUSE}}$ LOW Delay (Note 5)		150		100		ns
TCSR	$\overline{\text{CS}}$ to $\overline{\text{RD}}$ LOW Set-up Time		0		0		ns
TCSW	$\overline{\text{CS}}$ LOW to $\overline{\text{WR}}$ LOW Set-up Time		0		0		ns
TCY	Clock Period		480	5000	320	2000	ns
TDW	Data Valid to $\overline{\text{WR}}$ HIGH Delay		150		100		ns
TEAE	$\overline{\text{EACK}}$ LOW to END LOW Delay			200		175	ns
TEHPHR	END HIGH to $\overline{\text{PAUSE}}$ HIGH Data Read when Busy			5.5TCY+300		5.5TCY+200	ns
TEHPHW	END HIGH to $\overline{\text{PAUSE}}$ HIGH Write when Busy			200		175	ns
TEPW	END HIGH Pulse Width		400		300		ns
TEX	Execution Time		See Table 2				ns
TOP	Data Bus Output Valid to $\overline{\text{PAUSE}}$ HIGH Delay		0		0		ns
TPPWR	$\overline{\text{PAUSE}}$ LOW Pulse Width Read	Data	3.5TCY+50	5.5TCY+300	3.5TCY+50	5.5TCY+200	ns
		Status	1.5TCY+50	3.5TCY+300	1.5TCY+50	3.5TCY+200	
TPPWRB	END HIGH to $\overline{\text{PAUSE}}$ HIGH Read when Busy	Data	See Table 2				ns
		Status	1.5TCY+50	3.5TCY+300	1.5TCY+50	3.5TCY+200	
TPPWW	$\overline{\text{PAUSE}}$ LOW Pulse Width Write when Not Busy			TCSW+50		TCSW+50	ns
TPPWWB	$\overline{\text{PAUSE}}$ LOW Pulse Width Write when Busy		See Table 2				ns
TPR	$\overline{\text{PAUSE}}$ HIGH to Read HIGH Hold Time		0		0		ns
TPW	$\overline{\text{PAUSE}}$ HIGH to Write HIGH Hold Time		0		0		ns
TRCD	$\overline{\text{RD}}$ HIGH to C/ $\overline{\text{D}}$ Hold Time		0		0		ns
TRCS	$\overline{\text{RD}}$ HIGH to $\overline{\text{CS}}$ HIGH Hold Time		0		0		ns
TRO	$\overline{\text{RD}}$ LOW to Data Bus On Delay		50		50		ns
TRZ	$\overline{\text{RD}}$ HIGH to Data Bus Off Delay		50	200	50	150	ns
TSAPW	SVACK LOW Pulse Width		100		75		ns
TSAR	SVACK LOW to SVREQ LOW Delay			300		200	ns
TWCD	$\overline{\text{WR}}$ HIGH to C/D Hold Time		60		30		ns
TWCS	$\overline{\text{WR}}$ HIGH to CS HIGH Hold Time		60		30		ns
TWD	$\overline{\text{WR}}$ HIGH to Data Bus Hold Time		20		20		ns

NOTES:

- Typical values are for $T_A = 25^\circ\text{C}$, nominal supply voltages and nominal processing parameters.
- Switching parameters are listed in alphabetical order.
- Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF and timing reference levels of 0.8V and 2.0V.
- END HIGH pulse width is specified for $\overline{\text{EACK}}$ tied to VSS. Otherwise TEAE applies.
- $\overline{\text{PAUSE}}$ is pulled low for both command and data operations.
- TEX is the execution time of the current command (see the Command Execution Times table).
- $\overline{\text{PAUSE}}$ will go low at this point if $\overline{\text{CS}}$ is low and $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are high.

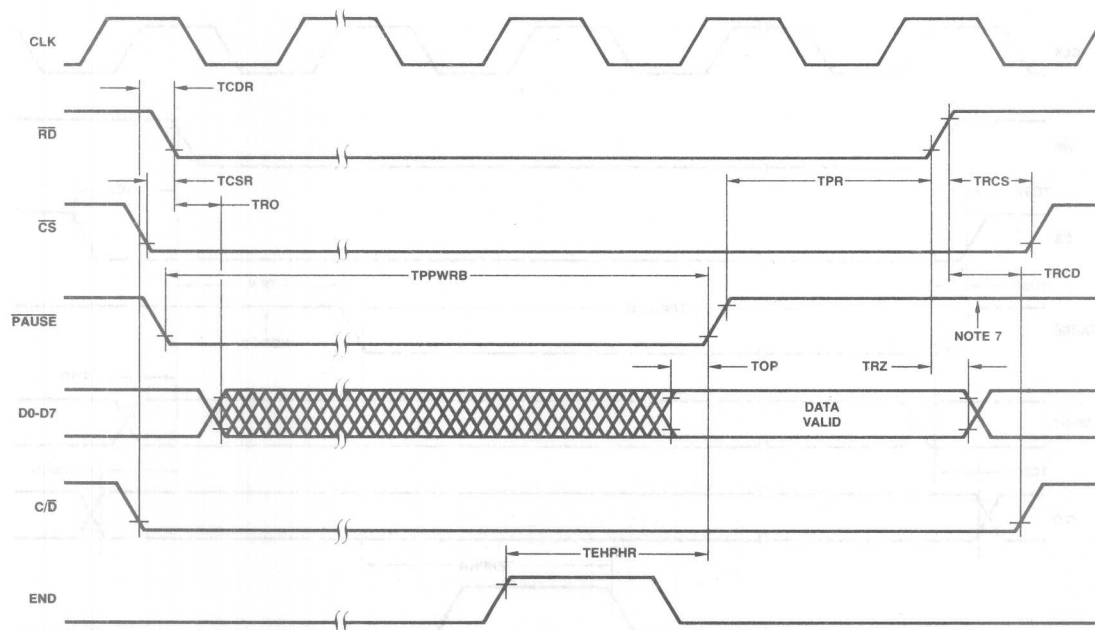
TIMING DIAGRAMS

READ OPERATION



MOS-208

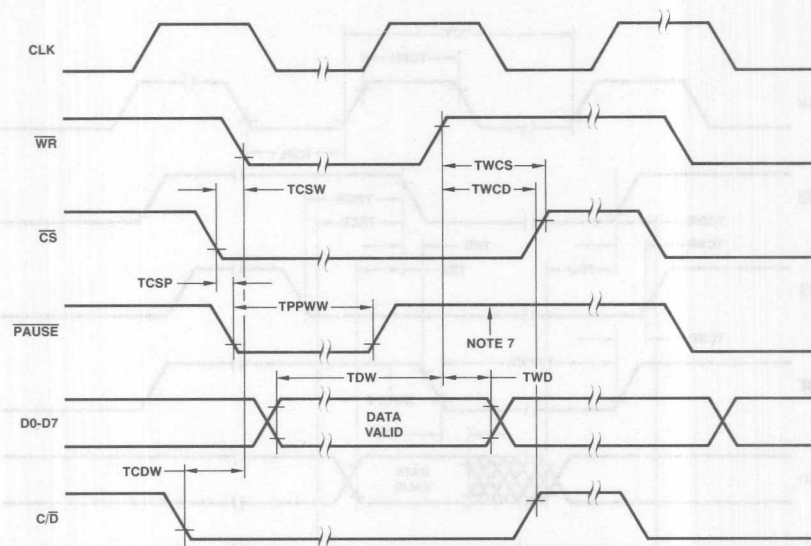
OPERAND READ WHEN Am9512 IS BUSY



MOS-209

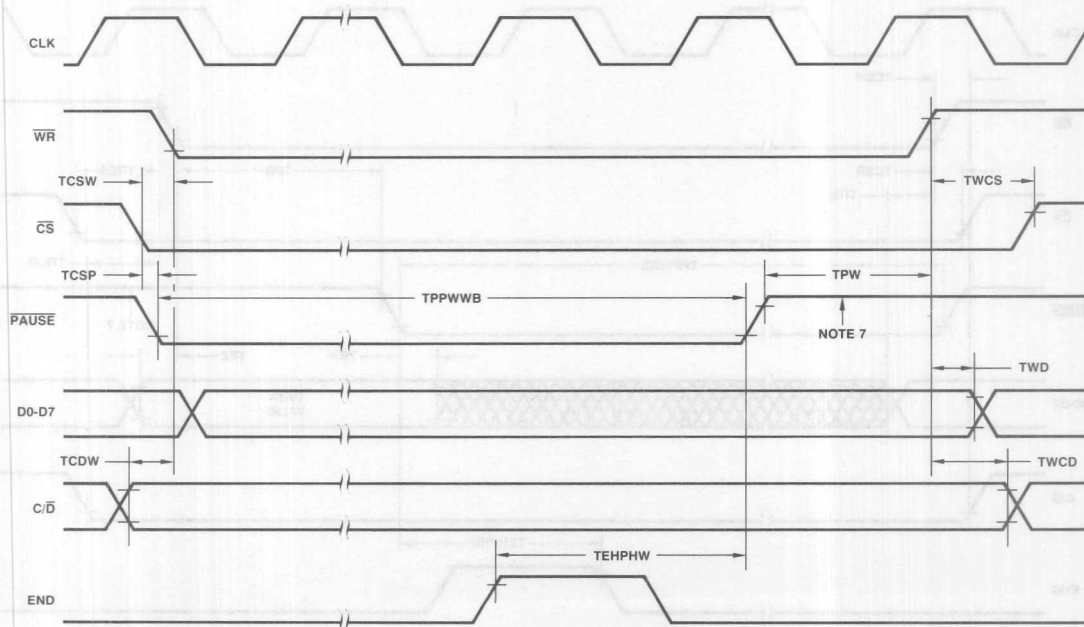
TIMING DIAGRAMS (Cont.)

OPERAND ENTRY



MOS-210

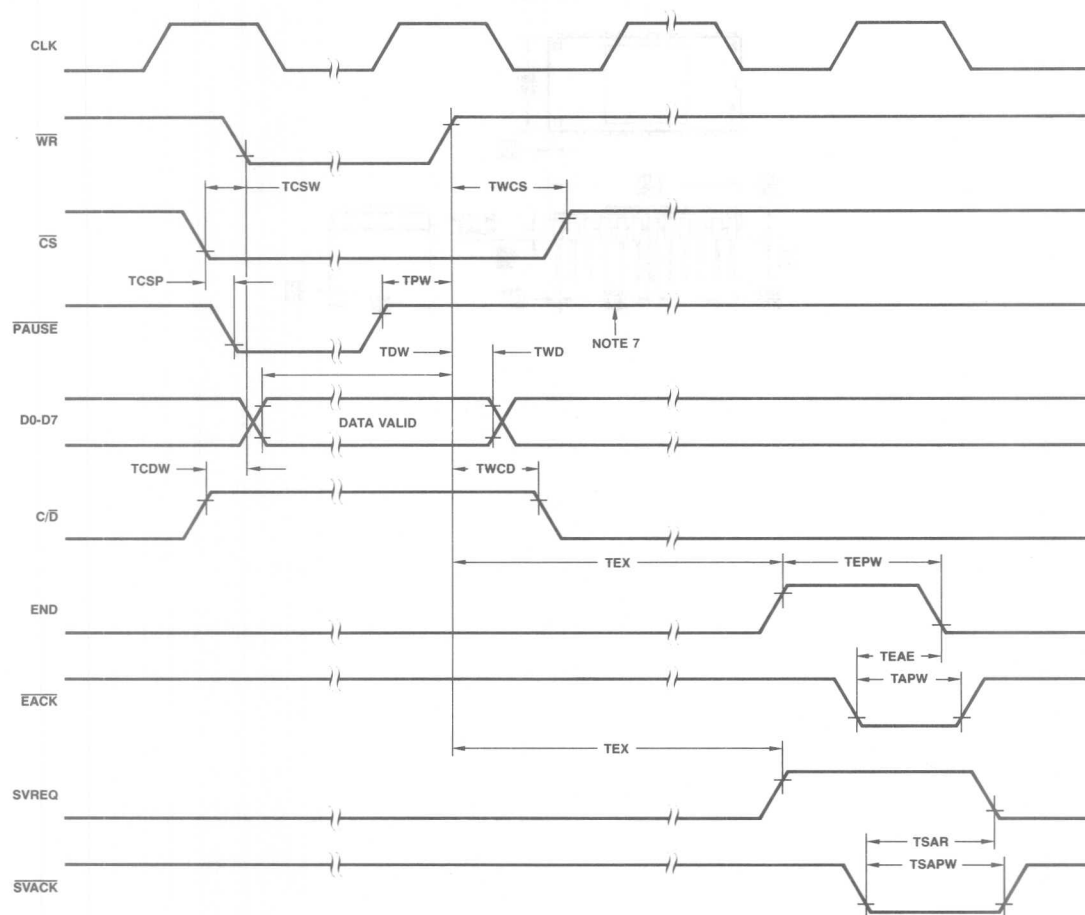
COMMAND OR DATA WRITE WHEN Am9512 IS BUSY



MOS-211

TIMING DIAGRAMS (Cont.)

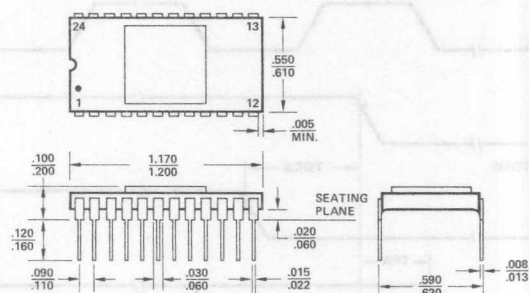
COMMAND INITIATION



MOS-212

PHYSICAL DIMENSIONS
Dual-In-Line

24-Pin Side-Brazed



Am9513

System Timing Controller

DISTINCTIVE CHARACTERISTICS

- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package
- 100% MIL-STD-883 reliability assurance testing

GENERAL DESCRIPTION

The Am9513 System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513 to be personalized for particular applications as well as dynamically reconfigured under program control.

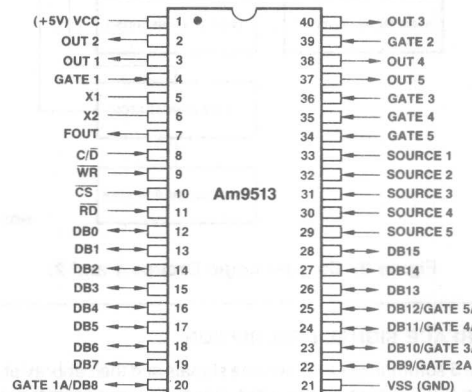
The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

TABLE OF CONTENTS

General Description	4-47
Pinout	4-47
Block Diagram	4-48
Interface Signal Description	4-48
Functional Description	4-50
Control Port Registers	4-51
Data Port Registers	4-53
Register Access	4-54
Master Mode Control Options	4-55
Operating Mode Description	4-57
Counter Mode Control Options	4-61
Command Descriptions	4-63
Maximum Ratings	4-67
Electrical Characteristics	4-67
Switching Characteristics	4-69
Applications Information	4-71

Advanced Micro Devices reserves the right to modify information contained in this document without notice.

CONNECTION DIAGRAM



Top View

Pin 1 is marked for orientation.

MOS-172

Figure 1.

ORDERING INFORMATION

Package Type	Temperature Range	Counting Frequency	
		7MHz	
Molded	0°C ≤ T _A ≤ +70°C	AM9513PC	
Hermetic*		AM9513DC	
		AM9513CC	
Hermetic	−55°C ≤ T _A ≤ +125°C	AM9513DM	

*Hermetic = Ceramic = DC = CC = D-40-1.

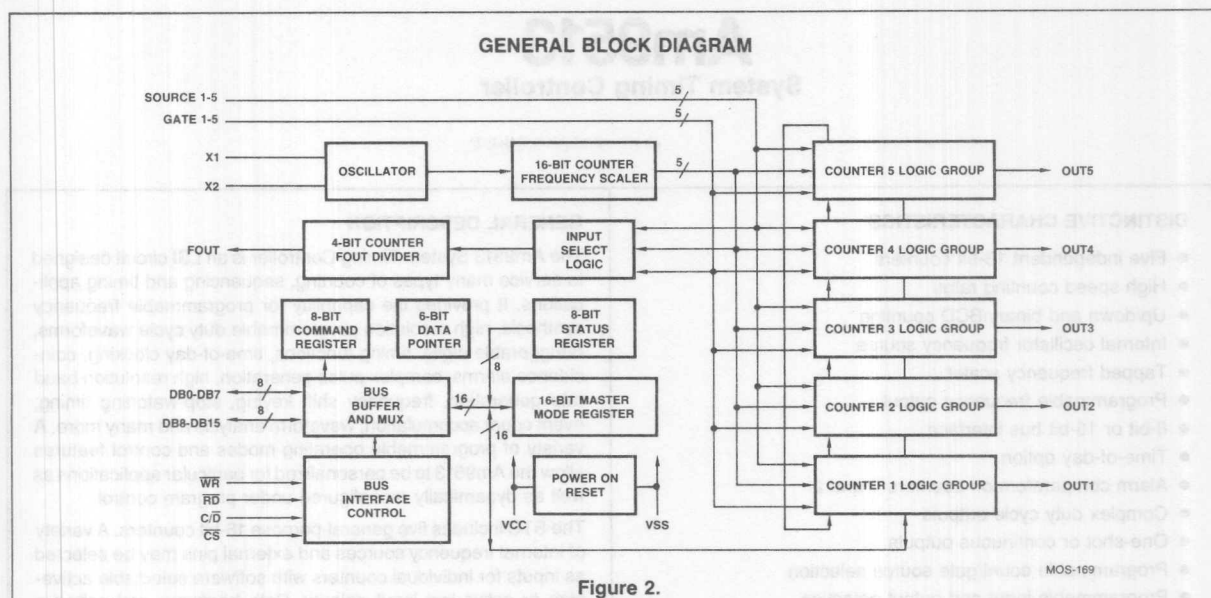


Figure 2.

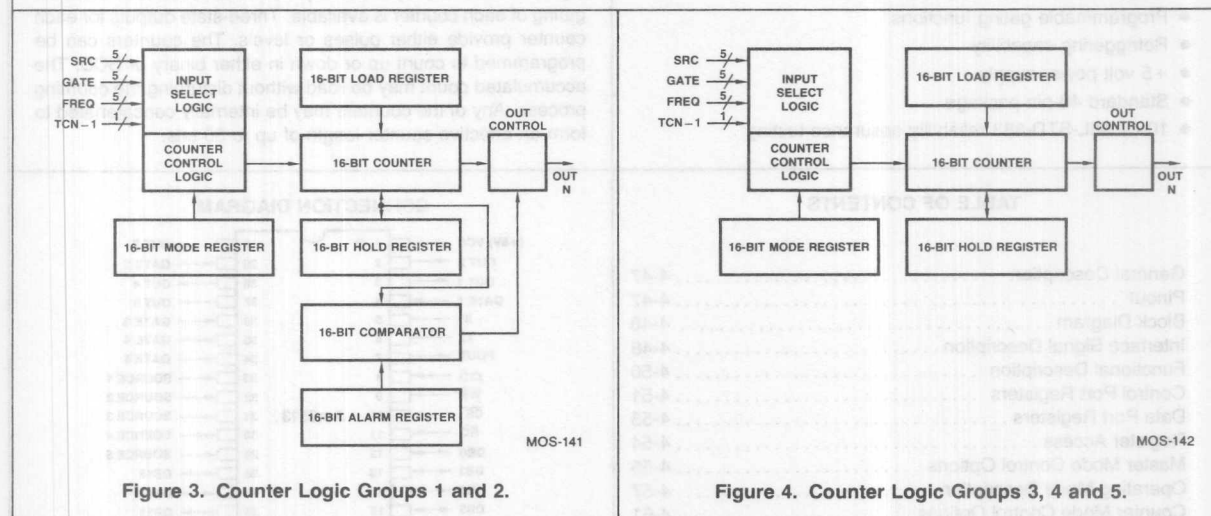


Figure 3. Counter Logic Groups 1 and 2.

Figure 4. Counter Logic Groups 3, 4 and 5.

INTERFACE SIGNAL DESCRIPTION

Figure 5 summarizes the interface signals and their abbreviations for the STC. Figure 1 shows the signal pin assignments for the standard 40-pin dual in-line package.

VCC: +5 volt power supply

VSS: Ground

X1, X2 (Crystal)

X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.

FOUT (Frequency Out, Output)

The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator.

GATE1-GATE5 (Gate, Inputs)

The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating

modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.

SRC1-SRC5 (Source, Inputs)

The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.

OUT1-OUT5 (Counter, Outputs)

Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.

DB0-DB7, DB8-DB15 (Data Bus, Input/Output)

The 16, bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when \overline{WR} and \overline{CS} are active and as outputs when \overline{RD} and \overline{CS} are active. When \overline{CS} is inactive, these pins are placed in a high-impedance state.

After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in

the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13-DB15 lines at a logic high level. Thereafter all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position.

When operating in the 8-bit data bus environment, DB8-DB15 will never be driven active by the Am9513. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 6). If unused they should be held high. When pulled low, a GATENA signal will disable the action of the corresponding counter N gating. DB13-DB15 should be held high in 8-bit bus mode whenever CS and WR are simultaneously active.

\overline{CS} (Chip Select, Input)

The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is high, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry.

\overline{RD} (Read, Input)

The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. \overline{WR} and \overline{RD} should be mutually exclusive.

\overline{WR} (Write, Input)

The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. \overline{WR} and \overline{RD} should be mutually exclusive.

C/\overline{D} (Control/Data, Input)

The Control/Data signal selects source and destination locations for read and write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing at the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	I/O, I	2
Read	\overline{RD}	Input	1
Write	\overline{WR}	Input	1
Chip Select	\overline{CS}	Input	1
Control/Data	C/\overline{D}	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 5. Interface Signal Summary.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 6. Data Bus Assignments.

FUNCTIONAL DESCRIPTION

The Am9513 block diagrams (Figures 2, 3 and 4) indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8 or 16 bits wide; in the 8-bit mode the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a control port and a data port. The control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the data port addressing.

Among the registers accessible through the data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16 bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and

comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input, and allows a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

A powerful command structure simplifies user interaction with the counters. A counter must be armed by one of the ARM commands before counting can commence. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. The ARM and DISARM commands permit software gating of the count process in some modes.

The LOAD command causes the counter to be reloaded with the value in either the associated Load register or the associated Hold register. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

The DISARM command disables further counting independent of any hardware gating. A disarmed counter may be reloaded using the LOAD command, may be incremented or decremented using the STEP command and may be read using the SAVE command. A count process may be resumed using an ARM command.

The SAVE command transfers the contents of a counter to its associated Hold register. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Two combinations of the basic commands exist to either LOAD AND ARM or to DISARM AND SAVE any combination of counters. Additional commands are provided to: step an individual counter by one count; set and clear an output toggle; issue a software reset; clear and set special bits in the Master Mode register; and load the Data Pointer register.

DB0	DB0	15
DB1	DB1	14
DB2	DB2	13
DB3	DB3	12
DB4	DB4	11
DB5	DB5	10
DB6	DB6	9
DB7	DB7	8
DATA 14	DATA 14	7
DATA 13	DATA 13	6
DATA 12	DATA 12	5
DATA 11	DATA 11	4
DATA 10	DATA 10	3
DATA 9	DATA 9	2
DATA 8	DATA 8	1
DATA 7	DATA 7	0

Pin	Type	Abbreviation	Signal
1	Power	VDD	+5 Volt
2	Power	VSS	Ground
3	IO	X1	Crystal
4	Input	RD	Read
5	Input	WR	Write
6	Input	CS	Chip Select
7	Input	CS	Chip Select
8	Input	CS	Chip Select
9	Input	CS	Chip Select
10	Input	CS	Chip Select
11	Input	CS	Chip Select
12	Input	CS	Chip Select
13	Input	CS	Chip Select
14	Input	CS	Chip Select
15	Input	CS	Chip Select
16	Input	CS	Chip Select
17	Output	FOUT	Frequency Out
18	Output	OUT	Out

CONTROL PORT REGISTERS

The STC is addressed by the external system as only two locations: a Control port and a Data port. Transfers at the Control port ($C/\bar{D} = \text{High}$) allow direct access to the command register when writing and the status register when reading. All other available internal locations are accessed for both reading and writing via the Data port ($C/\bar{D} = \text{Low}$). Data port transfers are executed to and from the location currently addressed by the Data Pointer register. Options available in the Master Mode register and the Data Pointer control structure allow several types of transfer sequencing to be used. See Figure 7.

Transfers to and from the control port are always 8 bits wide. Each access to the Control port will transfer data between the Command register (writes) or Status register (reads) and Data Bus DB0-DB7, regardless of whether the Am9513 is in 8- or 16-bit bus mode. When the Am9513 is in 8-bit bus mode, Data Bus pins DB13-DB15 should be held at a logic high whenever \overline{CS} and \overline{WR} are both active.

Command Register

The Command register provides direct control over each of the five general counters and controls access through the Data port by allowing the user to update the Data Pointer register. The "Command Description" section of this data sheet explains the detailed operation of each command. A summary of all commands appears in Figure 21. Six of the command types are used for direct software control of the counting process. Each of these six commands contains a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of the five general counters ($S1 = \text{Counter 1}$, $S2 = \text{Counter 2}$, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter.

Data Pointer Register

The 6-bit Data Pointer register is loaded by issuing the appropriate command through the control port to the Command register. As shown in Figure 7, the contents of the Data Pointer register are used to control the Data Port multiplexer, selecting which internal register is to be accessible through the Data Port.

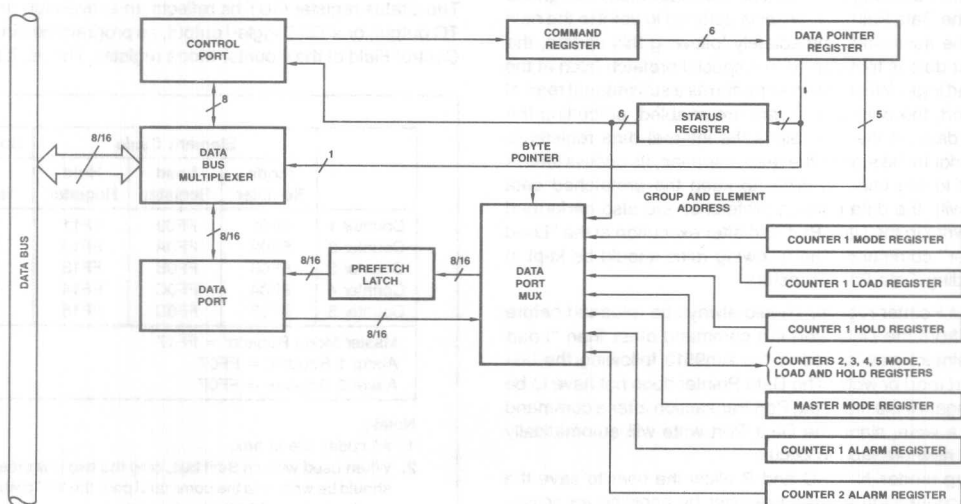
The Data Pointer consists of a 3-bit Group Pointer, a 2-bit Element Pointer and a 1-bit Byte Pointer, depicted in Figure 8. The Byte Pointer bit indicates which byte of a 16-bit register is to be transferred on the next access through the data port. Whenever the Data Pointer is loaded, the Byte Pointer bit is set to one, indicating a least-significant byte is expected. The Byte Pointer toggles following each 8-bit data transfer with an 8-bit data bus ($MM13 = 0$), or it always remains set with the 16-bit data bus option ($MM13 = 1$). The Element and Group pointers are used to select which internal register is to be accessible through the Data Port. Although the contents of the Element and Group Pointer in the Data Pointer register cannot be read by the host processor, the Byte Pointer is available as a bit in the Status register.

Random access to any available internal data location can be accomplished by simply loading the Data Pointer using the command shown in Figure 9 and then initiating a data read or data write. This procedure can be used at any time, regardless of the setting of the Data Pointer Control bit ($MM14$). When the 8-bit data bus configuration is being used ($MM13 = 0$), two bytes of data would normally be transferred following the issuing of the "Load Data Pointer" command.

To permit the host processor to rapidly access the various internal registers, automatic sequencing of the Data Pointer is provided. Sequencing is enabled by clearing Master Mode bit 14 ($MM14$) to zero. As shown in Figure 10, several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

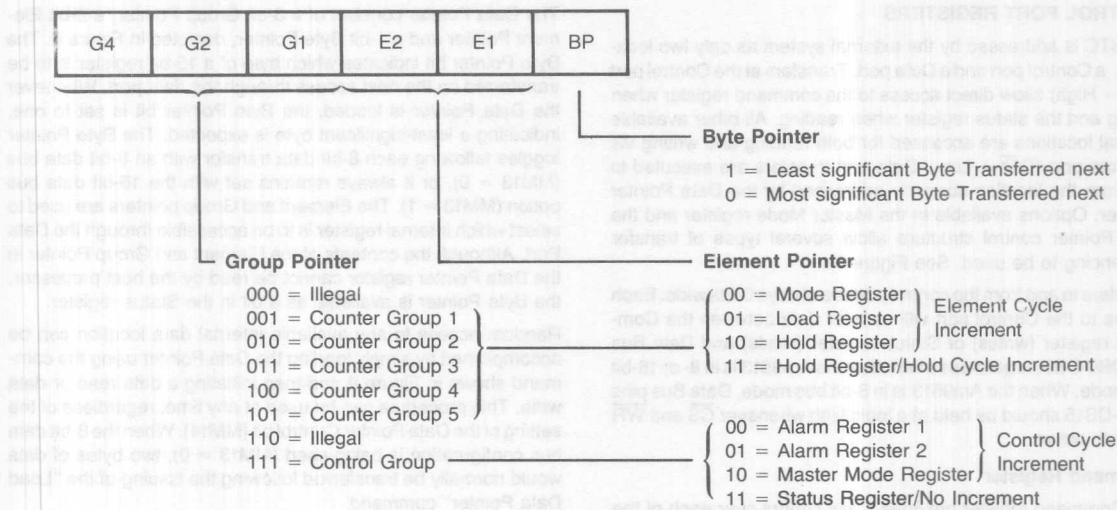
When $E1 = 0$ or $E2 = 0$ and $G4, G2, G1$ point to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If $E2, E1 = 11$ and a Counter Group is selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control



MOS-501

Figure 7. Am9513 Register Access.



MOS-173

Figure 8. Data Pointer Register.

cycle. If G4, G2, G1 = 111 and E2, E1 \neq 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the data port. Note that the Status register can also always be read directly through the Control port.

For all of these auto-sequence modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group Fields are incremented.

Prefetch Circuit

In order to minimize the read access time to internal Am9513 registers, a prefetch circuit is used for all read operations through the Data Port. Following each read or write operation through the Data Port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data Port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. In order to keep the prefetched data consistent with the data pointer, prefetches are also performed after each write to the Data Port and after execution at the "Load Data Pointer" command. The following rules should be kept in mind regarding Data Port Transfers.

1. The Data Pointer register should always be reloaded before reading from the Data Port if a command other than "Load Data Pointer" was issued to the Am9513 following the last Data Port read or write. The Data Pointer does not have to be loaded again if the first Data Port transaction after a command entry is a write, since the Data Port write will automatically cause a new prefetch to occur.
2. Operating modes N, O, Q and R allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold register. To

avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data Port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the OUT signal for each of the general counters. See Figures 11 and 19. The OUT signals reported are those internal to the chip after the polarity-select logic and just before the 3-state interface buffer circuitry.

The Status register OUT bit reflects an active-high or active-low TC output, or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D
Master Mode Register = FF17				
Alarm 1 Register = FF07				
Alarm 2 Register = FF0F				

Notes:

1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the 'FF' prefix should be used only for a 16-bit data bus interface.

Figure 9. Load Data Pointer Commands.

exact state of the OUT pin. When the Low Impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a High Impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active high if CM2 = 0 and active-low if CM2 = 1. When the High Impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the Low Impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the control port (see Figure 7) but may also be read via the data port as part of the Control Group.

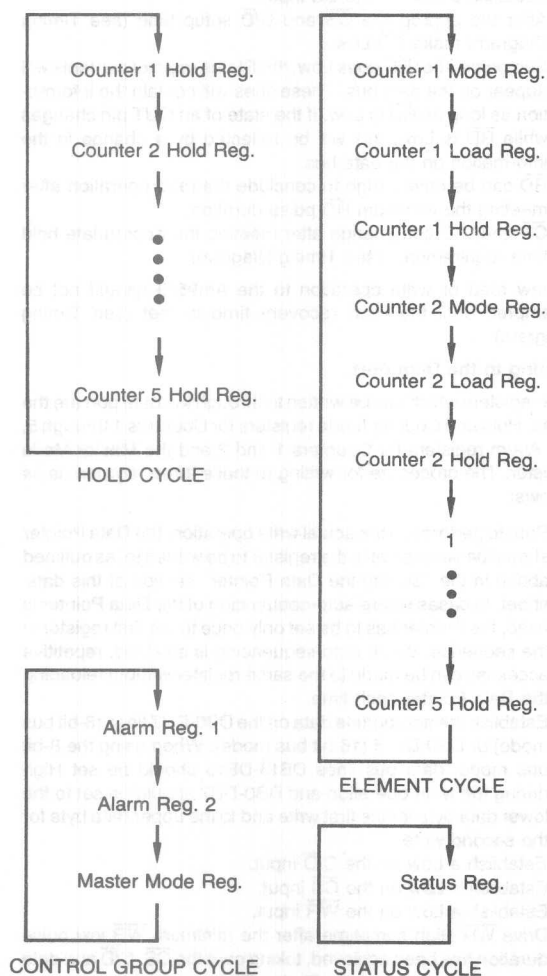


Figure 10. Data Pointer Sequencing.

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 3 and 4, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the data port. The counter itself is never directly accessed.

The 16-bit read/write Load register is used to control the effective period of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time that Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes the contents of either Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can be transparent by filling the Load register with all zeros.

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for modulo definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by the software SAVE command at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this data sheet describes the detailed control options available. Figure 18 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 3). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the

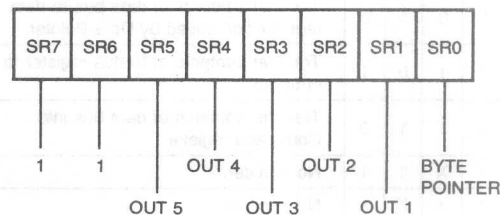


Figure 11. Status Register Bit Assignments.

comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

REGISTER ACCESS

Information Transfer Protocols

The control signal configurations for all information transfers on the Am9513 data bus are summarized in Figure 12. The interface control logic assumes these conventions:

1. \overline{RD} and \overline{WR} are never active at the same time.
2. \overline{RD} , \overline{WR} and $\overline{C/D}$ are ignored unless \overline{CS} is Low.

Command Initiation

The procedure for executing a command is as follows:

1. Establish the appropriate command on the DB0-DB7 lines. Figure 21 lists the command codes. When using the Am9513 in 16-bit mode, data bus lines DB8-DB15 should be set high during the write operation. In 8-bit data bus mode, DB13-DB15 should be set high during the write operation.
2. Establish a High on the $\overline{C/D}$ input.
3. Establish a Low on the \overline{CS} input.
4. Establish a Low on the \overline{WR} input.
5. Sometime after the minimum \overline{WR} low pulse duration has been achieved, drive \overline{WR} high, taking care the \overline{CS} , $\overline{C/D}$ and data setup times are met (see Timing Diagram).
6. After meeting the required \overline{CS} , $\overline{C/D}$ and data hold times, these signals can be changed (see Timing Diagram).

A new read or write operation to the Am9513 should not be performed until the write recovery time is met (see Timing Diagram).

Setting the Data Pointer Register

The Data Pointer register selects which register is to be accessed through the data port. The Pointer is set as follows:

1. Using Figures 8 and 9, select the appropriate Data Pointer Group and Element codes for the register to be accessed. Note that two codes are provided for the Hold registers, to accommodate both the Hold Cycle and Element Cycle auto-sequencing modes shown in Figure 10. If auto-sequencing is disabled, either Hold code may be used.

Signal Configuration				Data Bus Operation
\overline{CS}	$\overline{C/D}$	\overline{RD}	\overline{WR}	
0	0	0	1	Transfer contents of register addressed by Data Pointer to the data bus.
0	0	1	0	Transfer contents of data bus to data register addressed by Data Pointer.
0	1	0	1	Transfer contents of Status register to data bus.
0	1	1	0	Transfer contents of data bus into Command register.
X	X	1	1	No transfer.
1	X	X	X	No transfer.
X	X	0	0	Illegal Condition.

Figure 12. Data Bus Transfers.

2. Using the "Writing to the Command Register" procedure given above, write the appropriate "Load Data Pointer" command to the Command register. Note that the command summary in Figure 21 has the Group field and Element field switched from the format given in Figure 8.

The Data Pointer register is now set. Setting the Data Pointer register automatically sets the Byte Pointer to 1, indicating a least significant byte is expected for 8-bit data bus interfacing. If Master Mode register bit MM14 = 0, the Data Pointer will automatically sequence through one of the cycles shown in Figure 10 after reading or writing each register. For convenience, bit MM14 can be set or cleared by software command.

Reading the Status Register

The procedure for reading the Status register through the Control port is given in the following. The Status register can also be read from the data port as outlined in the Reading from the Data Port section of this data sheet.

1. Establish a High on the $\overline{C/D}$ input.
2. Establish a Low on the \overline{CS} input.
3. After the appropriate \overline{CS} and $\overline{C/D}$ setup time (see Timing Diagram) make \overline{RD} Low.
4. Sometime after \overline{RD} goes Low, the Status register contents will appear on the data bus. These lines will contain the information as long as \overline{RD} is Low. If the state of an OUT pin changes while \overline{RD} is Low, this will be reflected by a change in the information on the data bus.
5. \overline{RD} can be driven High to conclude the read operation after meeting the minimum \overline{RD} pulse duration.
6. \overline{CS} and $\overline{C/D}$ can change after meeting the appropriate hold time requirements (see Timing Diagram).

A new read or write operation to the Am9513 should not be attempted until the read recovery time is met (see Timing Diagram).

Writing to the Data Port

The registers which can be written to through the data port are the Load, Hold and Counter Mode registers for Counters 1 through 5, the Alarm registers for Counters 1 and 2 and the Master Mode register. The procedure for writing to these three registers is as follows:

1. Prior to performing the actual write operation, the Data Pointer should be set to point to the register to be written to, as outlined above in the "Setting the Data Pointer" section of this data sheet. In cases where auto-sequencing of the Data Pointer is used, the Pointer has to be set only once to the first register in the sequence. When auto-sequencing is disabled, repetitive accesses can be made to the same register without reloading the Data Pointer each time.
2. Establish the appropriate data on the DB0-DB7 lines (8-bit bus mode) or DB0-DB15 (16-bit bus mode). When using the 8-bit bus mode, data bus lines DB13-DB15 should be set High during the write operation and DB0-DB7 should be set to the lower data byte for the first write and to the upper data byte for the second write.
3. Establish a Low on the $\overline{C/D}$ input.
4. Establish a Low on the \overline{CS} input.
5. Establish a Low on the \overline{WR} input.
6. Drive \overline{WR} High sometime after the minimum \overline{WR} low pulse duration has been achieved, taking care the \overline{CS} , $\overline{C/D}$ and data setup times are met (see Timing Diagram).
7. After meeting the required \overline{CS} , $\overline{C/D}$ and data hold times, these signals can be changed (see Timing Diagram).
8. After meeting the write recovery time (see Timing Diagram) a new read or write operation can be performed. For the 8-bit bus mode, steps 2 through 7 should be repeated, this time

placing the high data byte on pins DB0-DB7. The user is not required to drive \overline{CS} or C/\overline{D} High between successive reads or writes, although this is permissible.

Reading From the Data Port

The registers which can be read from the Data port are the Load, Hold and Counter Mode registers for Counters 1 through 5, the Alarm registers for Counters 1 and 2, the Master Mode register and the Status register. The Status register can also be read from the Control port. The procedure for reading these registers is as follows:

1. Prior to performing the actual read operation, the Data Pointer should be set to point to the register to be read, as outlined in the "Settling the Data Pointer" section of this data sheet. In cases where auto-sequencing of the Data Pointer is used, the Pointer has to be set only once to the first register in the sequence. When auto-sequencing is disabled, repetitive accesses can be made to the same register without reloading the Data Pointer each time. Special care must be taken to reset the Data Pointer after issuing a command other than "Load Data Pointer" to the Am9513 or when operating a counter in modes N, O, Q or R. See the "Prefetch Circuit" section of this document for elaboration.
2. Establish a Low on the C/\overline{D} input.
3. Establish a Low on the \overline{CS} input.
4. Establish a Low on \overline{RD} after waiting for the appropriate \overline{CS} and C/\overline{D} setup time (see Timing Diagram).
5. Sometime after \overline{RD} goes Low, the register contents will appear on the data bus. In both 8- and 16-bit bus modes the low register byte will appear on DB0-DB7. In addition, in 16-bit bus mode, the upper register byte will appear on DB8-DB15. For 8-bit bus mode, pins DB8-DB15 are not driven by the Am9513.

This information will remain stable as long as \overline{RD} is Low. If the register value is changed during the read, the change will not be reflected by a change in the data being read, for the reasons outlined in the "Prefetch Circuit" section of this document.

6. \overline{RD} can be driven High to conclude the read operation after meeting the minimum \overline{RD} pulse duration.
7. \overline{CS} and C/\overline{D} can change after meeting appropriate hold time requirements (see Timing Diagram).
8. After waiting the minimum read recovery time (see Timing Diagram), a new read or write operation can be started. For 8-bit bus mode, steps 2 through 7 should be repeated to read out the high register byte on DB0-DB7. (If the Status register is being read in 8-bit mode, the two reads will return the Status register each time. In 16-bit mode, reads from the Status register return undefined data on DB8-DB15.) The user is not required to drive \overline{CS} or C/\overline{D} High between successive reads or writes, although this is permissible.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, time-of-day operation, comparator controls, data bus width and data pointer sequencing. Figure 13 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition they can all be changed by writing directly to the Master Mode register.

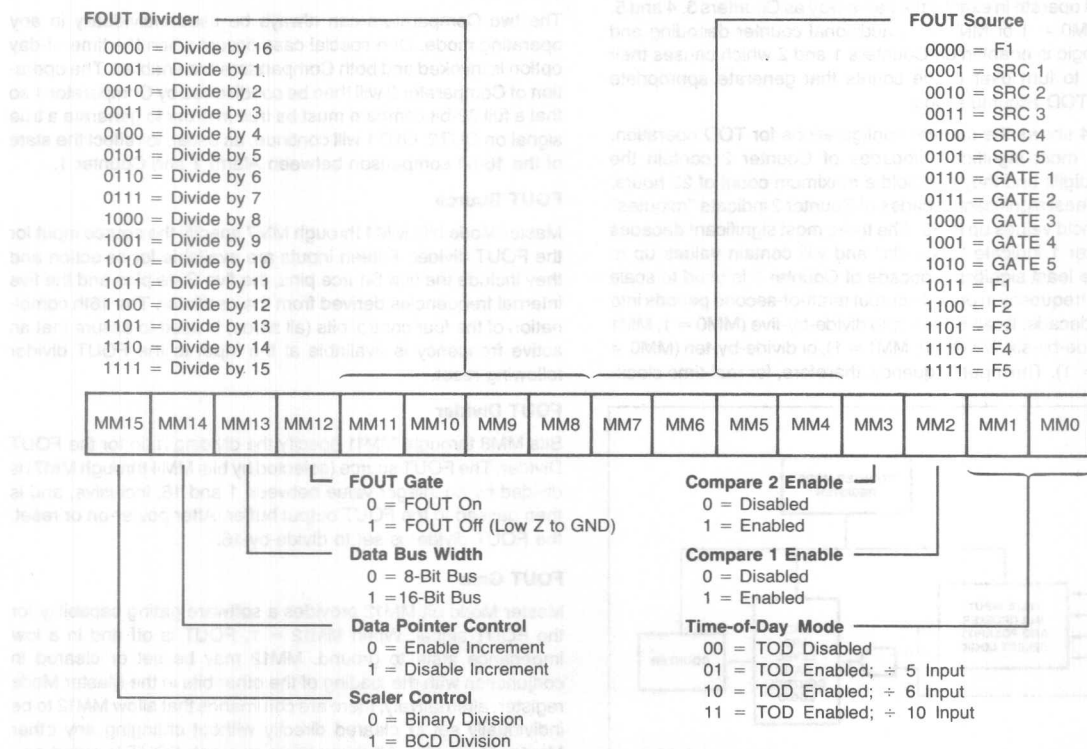


Figure 13. Master Mode Register Bit Assignments.

MOS-180

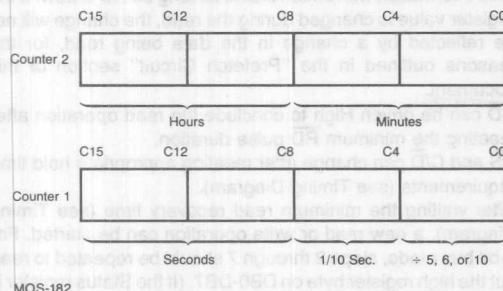


Figure 14. Time-of-Day Storage Configuration.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

- Time-of-day disabled
- Both Comparators disabled
- FOUT Source is frequency F1
- FOUT Divider set for divide-by-16
- FOUT gated on
- Data Bus 8 bits wide
- Data Pointer Sequencing enabled
- Frequency Scaler divides in binary

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the time-of-day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2 which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations.

Figure 14 shows the counter configurations for TOD operation. The two most significant decades of Counter 2 contain the "hours" digits and they can hold a maximum count of 23 hours. The two least significant decades of Counter 2 indicate "minutes" and will hold values up to 59. The three most significant decades of Counter 1 indicate "seconds" and will contain values up to 59.9. The least significant decade of Counter 1 is used to scale the input frequency in order to output tenth-of-second periods into the next decade. It can be set up to divide-by-five (MM0 = 1, MM1 = 0), divide-by-six (MM0 = 0, MM1 = 1), or divide-by-ten (MM0 = 1, MM1 = 1). The input frequency, therefore, for real-time clock-

ing can be, respectively, 50Hz, 60Hz or 100Hz. With divide-by-ten specified and with 100Hz input, the least significant decade of Counter 1 accumulates time in hundredths of seconds (tens of milliseconds). For accelerated time applications other input frequencies may be useful.

The input for Counter 2 should be the TC output of Counter 1, connected either internally or externally, for TOD operation. Both counters should be set up for BCD counting. The Load registers should be used to initialize the counters to the proper time. Either count up or count down may be used.

To read the time, a SAVE command should be issued to Counters 1 and 2. Because counts ripple between the counters, the possibility exists of the SAVE command occurring after Counter 1 has counted but before Counter 2 has. This would result in an incorrectly saved time. To guard against this, Counter 2 should be resaved if Counter 1's saved value indicates a ripple carry/borrow may have been generated. In other words, Counter 2 should be resaved if the value saved from Counter 1 is 0 (up counting), 59.94 (down counting, MM1-MM0 = 01), 59.95 (down counting, MM1-MM0 = 10), or 59.99 (down counting, MM1-MM0 = 11). By the time this test is performed and Counter 2 is resaved, any rippling carry/borrow will have updated Counter 2.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counter 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the time-of-day option is invoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

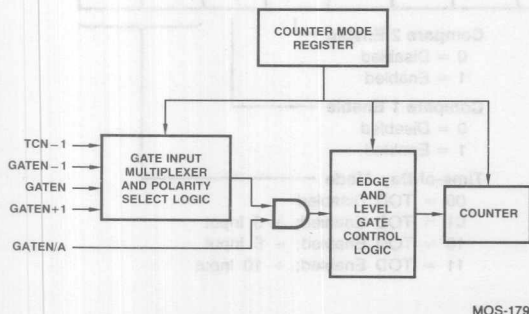


Figure 15. Gating Control.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16 bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513 is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs, and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 15. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

Thus the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides

the oscillator frequency in binary steps so that each sub-frequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 16).

OPERATING MODE DESCRIPTIONS

Counter Mode register bits CM15-CM13 and CM7-CM5 select the operating mode for each counter (see Figure 17). To simplify references to a particular mode, each mode is assigned a letter from A through X.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes states that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this data sheet, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

Mode A is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

Mode B is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive.

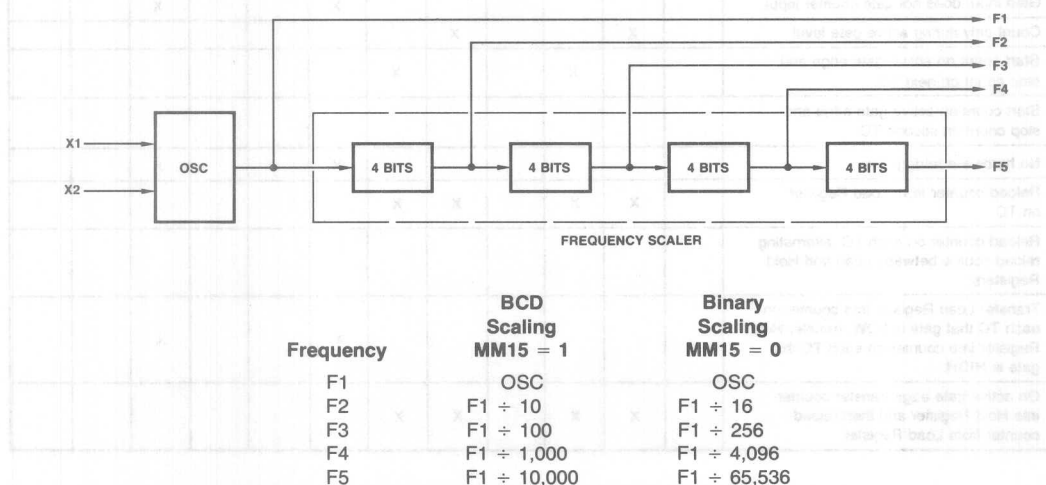


Figure 16. Frequency Scaler Ratios.

MOS-150

Operating Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC.			X			X						
Start count on active gate edge and stop count on second TC.									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load Register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold Registers.							X	X	X	X	X	X
Transfer Load Register into counter on each TC that gate is LOW; transfer Hold Register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold Register and then reload counter from Load Register.												

Operating Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly					X	X				X		
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC.			X			X						
Start count on active gate edge and stop count on second TC.												
No hardware retriggering							X			X		
Reload counter from Load Register on TC		X	X		X	X						
Reload counter on each TC, alternating reload source between Load and Hold Registers.												
Transfer Load Register into counter on each TC that gate is LOW; transfer Hold Register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold Register and then reload counter from Load Register.		X	X		X	X						

Note: Operating modes M, P, T, U, W and X are reserved and should not be used.

Figure 17. Counter Control Interaction.

This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting until a new ARM command is issued.

MODE C **Hardware-Triggered Strobe**

Mode C is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.

MODE D **Rate Generator with No Hardware Gating**

Mode D is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC the counter will reload itself from the Load register; hence the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E **Rate Generator with Level Gating**

Mode E is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

MODE F **Non-Retriggerable One-Shot**

Mode F provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G **Software-Triggered Delayed Pulse One-Shot**

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration.

MODE H **Software-Triggered Delayed Pulse One-Shot with Hardware Gating**

Mode H is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is active and disregard those source edges that occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

MODE I **Hardware-Triggered Delayed Pulse Strobe**

Mode I is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.

MODE J **Variable Duty Cycle Rate Generator with No Hardware Gating**

Mode J will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

MODE K **Variable Duty Cycle Rate Generator with Level Gating**

Mode K is identical to Mode J except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the high and low portions of the output waveform.

MODE L **Hardware-Triggered Delayed Pulse One-Shot**

Mode L is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge;

Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

Mode N provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

Mode O is similar to Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge the Load register contents will be transferred into the counter. Counting will resume on the second source edge after a retrigger.

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

Mode Q provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while

the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the counter. Counting will resume on the second qualified source edge after the retriggering gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE R

Retriggerable One-Shot

Mode R is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application at a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.

MODE S

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle.

MODE V

Frequency-Shift Keying

Mode V provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is High, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 18 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

- Output low impedance to ground
- Count down
- Count binary
- Count once
- Load register selected
- No retriggering
- F1 input source selected
- Positive-true input polarity
- No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 19 shows a schematic representation of the output control logic. The OUT pin may be off and in a high impedance state, or it may be off with a low impedance to ground. The three remaining valid combinations represent the two basic output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 20 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 20 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state and TC will indicate the counter state that would have been zero had no parallel transfer occurred.

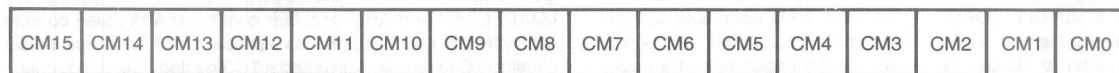
The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse.

Count Source Selection

- 0XXXX = Count on Rising Edge
- 1XXXX = Count on Falling Edge
- X0000 = TCN - 1
- X0001 = SRC 1
- X0010 = SRC 2
- X0011 = SRC 3
- X0100 = SRC 4
- X0101 = SRC 5
- X0110 = GATE 1
- X0111 = GATE 2
- X1000 = GATE 3
- X1001 = GATE 4
- X1010 = GATE 5
- X1011 = F1
- X1100 = F2
- X1101 = F3
- X1110 = F4
- X1111 = F5

Count Control

- 0XXXX = Disable Special Gate
- 1XXXX = Enable Special Gate
- X0XXX = Reload from Load
- X1XXX = Reload from Load or Hold
- XX0XX = Count Once
- XX1XX = Count Repetitively
- XXX0X = Binary Count
- XXX1X = BCD Count
- XXXX0 = Count Down
- XXXX1 = Count Up



Gating Control

- 000 = No Gating
- 001 = Active High Level TCN - 1
- 010 = Active High Level GATE N + 1
- 011 = Active High Level GATE N - 1
- 100 = Active High Level GATE N
- 101 = Active Low Level GATE N
- 110 = Active High Edge GATE N
- 111 = Active Low Edge GATE N

Output Control

- 000 = Inactive, Output Low
- 001 = Active High Terminal Count Pulse
- 010 = TC Toggled
- 011 = Illegal
- 100 = Inactive, Output High Impedance
- 101 = Active Low Terminal Count Pulse
- 110 = Illegal
- 111 = Illegal

Note: See Figure 17 for restrictions on Count Control and Gating Control bit combinations.

MOS-176

Figure 18. Counter Mode Register Bit Assignments.

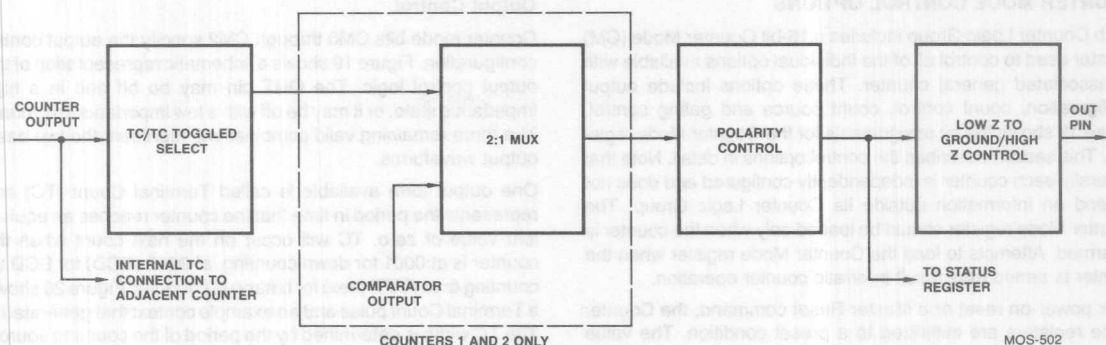


Figure 19. Output Control Logic.

The toggle output is 1/2 the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criteria of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands.

TC (TERMINAL COUNT)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD-and-ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count

source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD-and-ARM command (see 2 below) or by the application of a STEP command.

2. If a LOAD or LOAD-and-ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD-and-ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC, and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD-and-ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

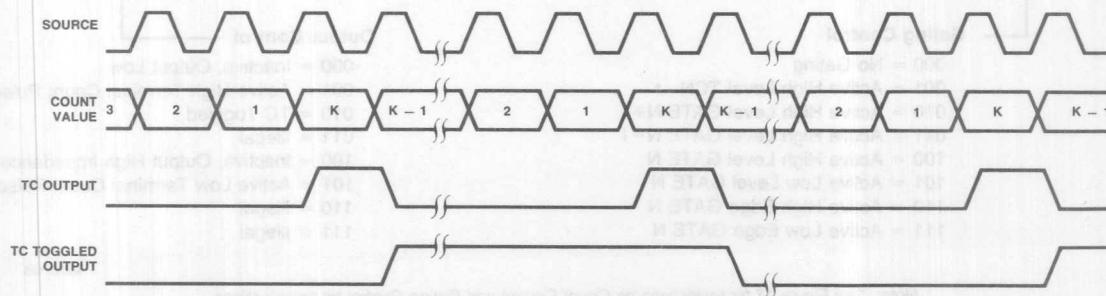


Figure 20. Counter Output Waveforms.

MOS-503

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register or the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a GATE pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycle ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend on the status of the Gating Control field and bits CM5 and CM6.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0 hardware retriggering does not occur; when CM7 = 1 the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

Whenever hardware retriggering is enabled (Modes N, O, Q and R) all active going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but reload it.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0 the Gate input has no effect on the counting; when CM7 = 1 the GATE N input specifies the reload source (either the Load or Hold register) used to reload the counter when TC occurs. Figure 17 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five

of the available inputs are internal frequencies derived from the internal oscillator (see Figure 16 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80 bits long on one Am9513 chip. When TCN - 1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM13 through CM15 specify the hardware gating options. When "no gating" is selected (000) the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes low, counting is simply suspended until the Gate goes high again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC output from the adjacent lower-numbered counter as the gate. Thus, one counter may be configured to generate a counting "window" for another counter.

COMMAND DESCRIPTIONS

The command set for the Am9513 allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 7).

All available commands are described in the following text. Figure 21 summarizes the command codes and includes a brief description of each function. Figure 22 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of the five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set output bit N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear output bit N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	1	1	1	Master reset

Figure 21. Am9513 Command Summary.

occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commence. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge; execution of a LOAD or LOAD-and-ARM command; or execution of a STEP command.)

In modes which alternate reload sources (Modes G-L), the ARming operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following

each ARM or LOAD-and-ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

In edge gating modes (Modes C, F, I, L, O and R) after disarming and rearming a triggered counter, a new Gate edge will be required to resume counting. In Modes C, F, I and L counting will resume from the current counter value. In modes O and R the new Gate edge will both start and retrigger the counter, causing the counter to be reloaded with a new value.

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger, or as counter initialization prior to active hardware gating.

If a LOAD or LOAD-and-ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse, internal to the Am9513, and the Am9513 is expecting to go into TC on the next count pulse. The reload

source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD-and-ARM command while a counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the reload source used will be that to be used for the upcoming TC. (The LOADING operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G-L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter) the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD-and-ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A-C and N-O, and Modes G-I and S if the current TC is the second in the cycle) the ARM part of the LOAD-and-ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G-L) the ARming operating will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other control conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARming. This count may be generated by a source edge, by a LOAD or LOAD-and-ARM command (the LOAD-and-ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

$(001 \leq N \leq 101)$

Description: The output toggle for counter N is set. The OUTN signal will be driven high unless a TC output is specified.

Clear Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

$(001 \leq N \leq 101)$

Description: The output toggle for counter N is reset. The OUTN signal will be driven low unless a TC output is specified.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

$(001 \leq N \leq 101)$

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

$(G4, G2, G1 \neq 000, \neq 110)$

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer

register as shown in Figure 8. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing.

Enable 16-Bit Data Bus

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Master Reset

Coding:	C7	C6	C5	C4	C3	C2	C1	C0
	1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation is given in the following.

1. Using the procedure given in the "Command Initiation" section of this data sheet, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this data sheet, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 9.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
* 1	1	1	1	1	X	X	X

*Unused except when XXX = 111.

Figure 22. Am9513 Unused Command Codes.

MAXIMUM RATINGS beyond which useful life may be impaired

Storage Temperature	-65°C to +150°C
Ambient Temperature Under Bias	-55°C to +125°C
VCC with Respect to VSS	-0.5V to +7.0V
All Signal Voltages with Respect to VSS	-0.5V to +7.0V
Power Dissipation (Package Limitation)	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Temperature	VCC	VSS
Am9513DC	0°C ≤ T _A ≤ +70°C	+5V ±5%	0V
Am9513DM	-55°C ≤ T _A ≤ +125°C	+5V ±5%	0V

ELECTRICAL CHARACTERISTICS over operating range (Notes 1 and 2)

Parameter	Description	Test Conditions	Min	Typ	Max	Units
VIL	Input Low Voltage	All Inputs Except X2	VSS - 0.5		0.8	Volts
		X2 Input	VSS - 0.5		0.8	
VIH	Input High Voltage	All Inputs Except X2	2.0		VCC	Volts
		X2 Input	3.4		VCC	
VITH	Input Hysteresis (SRC and GATE Inputs Only)		0.2	0.3		Volts
VOL	Output Low Voltage	IOL = 3.2mA			0.4	Volts
VOH	Output High Voltage	IOH = -200μA	2.4			Volts
		IOH = -1.5mA	1.5			
IIX	Input Load Current (Except X2)	VSS ≤ VIN ≤ VCC			±10	μA
IOZ	Output Leakage Current (Except X1)	VSS ≤ VOUT ≤ VCC High Impedance State			±25	μA
ICC	VCC Supply Current	T _A = -55°C			275	mA
		T _A = 0°C			225	
		T _A = +25°C		160		
CIN	Input Capacitance	f = 1MHz, T _A = +25°C, All pins not under test at 0V.			10	pF
COUT	Output Capacitance				15	
CIO	IN/OUT Capacitance				20	

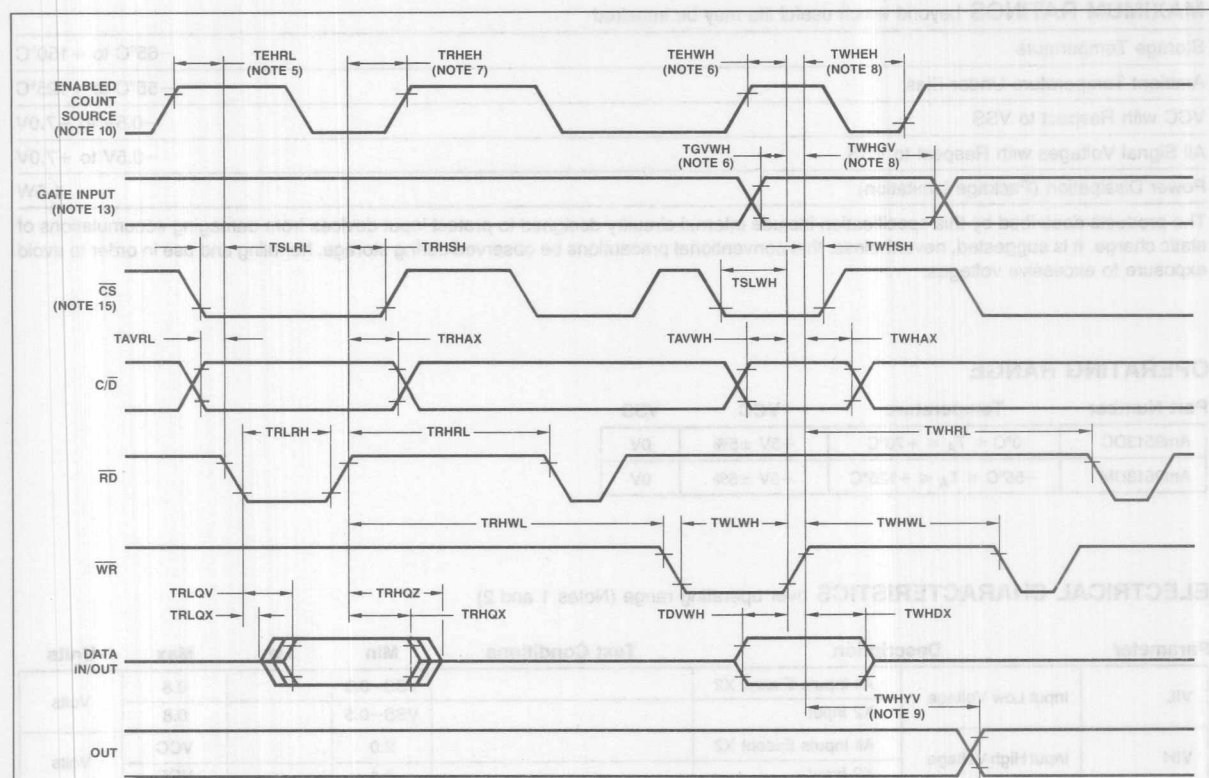


Figure 23. Bus Transfer Switching Waveforms.

MOS-183

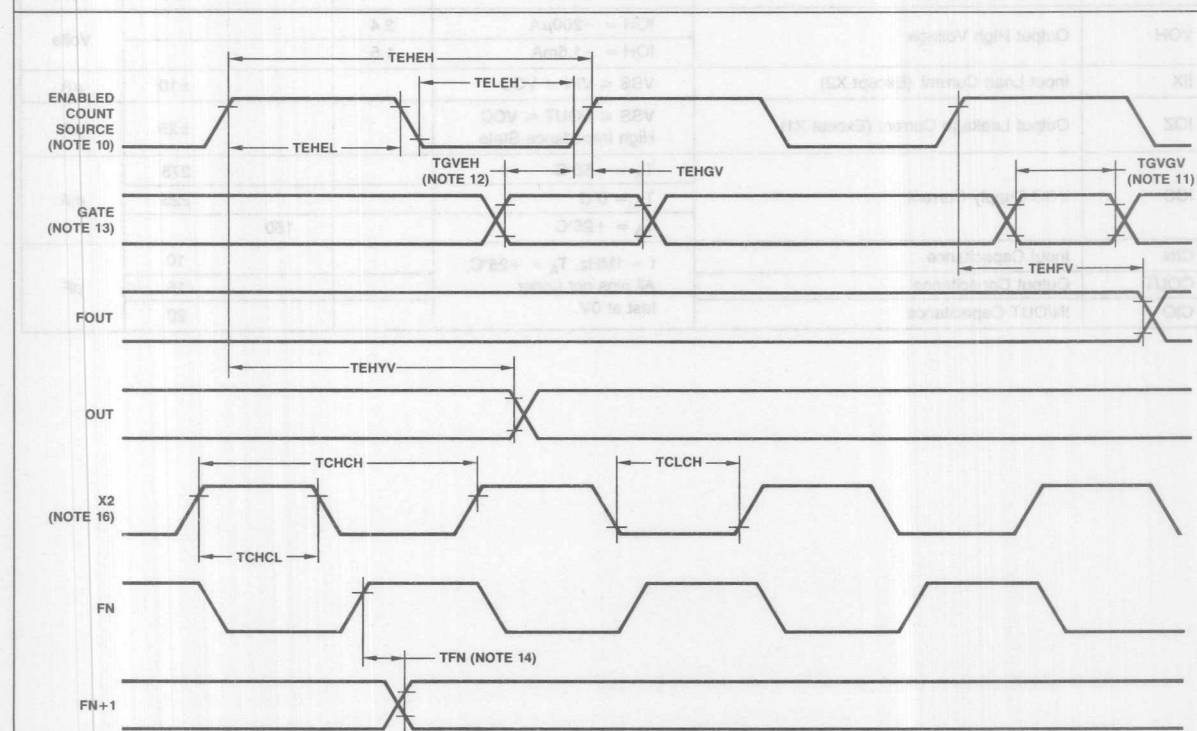


Figure 24. Counter Switching Waveforms.

MOS-184

SWITCHING CHARACTERISTICS over operating range (Notes 2, 3, 4)

		Am9513					
Parameter	Description	Figure	Min	Max	Min	Max	Units
TAVRL	C/D Valid to Read Low	23	25				ns
TAVWH	C/D Valid to Write High	23	170				ns
TCHCH	X2 High to X2 High (X2 Period)	24	145				ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width)	24	70				ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width)	24	70				ns
TDVWH	Data In Valid to Write High	23	80				ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 10)	24	145				ns
TEHEL TELEH	Count Source Pulse Duration (Note 10)	24	70				ns
TEHFV	Count Source High to FOUT Valid (Note 10)	24		500			ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 10, 12, 13)	24	40				ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 5, 10)	23	190				ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 6, 10)	23	100				ns
TEHYV	Count Source High to Out Valid (Note 10)	TC Output	24		300		ns
		Immediate or Delayed Toggle Output	24		300		
		Comparator Output	24		350		
TFN	FN High to FN+1 Valid (Note 14)	24		75			ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 10, 12, 13)	24	70				ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 11, 13)	24	145				ns
TGVWH	Gate Valid to Write High (Notes 6, 13)	23	0				ns
TRHAX	Read High to C/D Don't Care	23	0				ns
TRHEH	Read High to Count Source High (Notes 7, 10)	23	0				ns
TRHQX	Read High to Data Out Invalid	23	20				ns
TRHQZ	Read High to Data Out at High Impedance (Data Bus Release Time)	23		85			ns
TRHRL	Read High to Read Low (Read Recovery Time)	23		1000			ns
TRHSH	Read High to CS High (Note 15)	23	0				ns
TRHWL	Read High to Write Low (Read Recovery Time)	23		1000			ns
TRLQV	Read Low to Data Out Valid	23		160			ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)	23	20				ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 15)	23	160				ns
TSLRL	CS Low to Read Low (Note 15)	23	20				ns
TSLWH	CS Low to Write High (Note 15)	23	170				ns
TWHAX	Write High to C/D Don't Care	23	0				ns
TWHDX	Write High to Data In Don't Care	23	0				ns
TWHEH	Write High to Count Source High (Notes 8, 10, 17)	23	400				ns
TWHGV	Write High to Gate Valid (Notes 8, 13, 17)	23	400				ns
TWHRL	Write High to Read Low (Write Recovery Time)	23		1000			ns
TWHSH	Write High to CS High (Note 15)	23	0				ns
TWHWL	Write High to Write Low (Write Recovery Time)	23		1000			ns
TWHYV	Write High to Out Valid (Note 9, 17)	23		650			ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 15)	23	150				ns

NOTES:

1. Typical values are for $T_A = 25^\circ\text{C}$, nominal supply voltage and nominal processing parameters.
2. Test conditions assume transition times of 10ns or less, timing reference levels of 0.8V and 2.0V and output loading of one TTL gate plus 100pF, unless otherwise noted.
3. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

A (Address) = C/\overline{D}

C (Clock) = X2

D (Data In) = DB0-DB15

E (Enabled counter source input) = SRC1-SRC5, GATE1-GATE5, F1-F5, TCN-1

F = FOUT

G (Counter gate input) = GATE1-GATE5, TCN-1

Q (Data Out) = DB0-DB15

R (Read) = \overline{RD}

S (Chip Select) = \overline{CS}

W (Write) = \overline{WR}

Y (Output) = OUT1-OUT5

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = High

L = Low

V = Valid

X = unknown or don't care

Z = high impedance

4. Switching parameters are listed in alphabetical order.
5. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.
6. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write. Failure to meet this setup time when issuing commands to the counter may result in incorrect counter operation.

7. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.
8. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation. Failure to meet this hold time when issuing commands to the counter may result in incorrect counter operation.
9. This parameter applies to cases where the write operation causes a change in the output bit.
10. The enabled count source is one of F1-F5, TCN-1, SRC1-SRC5 or GATE1-GATE5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.
11. This parameter applies to edge gating (CM15-CM13 = 110 or 111) and gating when both CM7 = 1 and CM15-CM13 \neq 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.
12. This parameter applies to both edge and level gating (CM15-CM13 = 001 through 111) and gating when both CM7 = 1 and CM15-CM13 = 000. This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge. Failure to meet the required setup and hold times may result in incorrect counter operation.
13. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.
14. Signals F1-F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals.
15. This timing specification assumes that \overline{CS} is active whenever \overline{RD} or \overline{WR} are active. \overline{CS} may be held active indefinitely.
16. This parameter assumes X2 is driven from an external gate with a square wave.
17. This parameter assumes that the write operation is to the command register.

APPLICATION INFORMATION

The X1 and X2 inputs can be driven with a RC network, an external TTL-level square wave, or a crystal. Figure 25 shows the suggested methods of connecting different frequency sources to the internal oscillator input.

The use of a crystal provides a highly accurate frequency source at moderate cost, and accordingly, will usually be the preferred method of operation. The Am9513 is designed to use a crystal in a parallel-resonant mode. The two ceramic capacitors connecting X1 and X2 to ground ensure proper loading on the crystal. The capacitor to X2 may be an adjustable type for fine-tuning the resonant frequency for critical applications.

An RC network provides a very low cost frequency source but may exhibit large frequency variations over recommended power supply and temperature ranges. Note that there is a resistor internal to the Am9513 in parallel with any external resistance.

Initialization Procedures

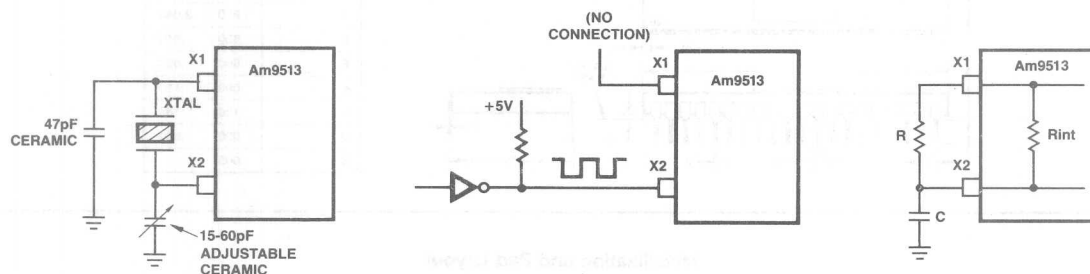
The reset function in the Am9513 is accomplished in two ways: automatically during power-up and by software Master Reset command. Power-up reset circuitry is internally triggered by the rising VCC voltage when a predetermined threshold is reached. An internal flip-flop is set by the rising supply voltage and controls the reset operation. The reset flip-flop remains set until cleared by the first active Chip Select input. A reset may also be initiated by the host processor by entering the Master Reset command. This software reset is active for the duration of the command write; otherwise it performs the same function as the power-up reset.

Following either type of Reset, all five counters are disabled, 0B00 is loaded into each Counter Mode register, and 0000 is loaded in the Master Mode register. This results in each counter being configured to count down in binary on the positive-going edge of the internal F1 frequency source with no repetition or gating. The Master Mode register is cleared to configure the Am9513 for an 8-bit data bus width; binary division of the internal oscillator; FOUT gated on and set to divide F1 by 16; time-of-day mode and comparators 1 and 2 disabled; and the Data Pointer increment enabled.

Reset will clear the Load and Hold registers for each counter but will not change either the counter contents or the Data Pointer register. Following a reset, the "Load All Counters" command (opcode 5F hex) should be issued to clear any counters that may be at TC. The Master Mode and Counter Mode, Load and Hold registers may now be set.

The following initialization procedure should be followed on Counters 1 and 2 when Time-of-Day mode is selected.

1. Set Time-of-Day enabled in the Master Mode register and load Counter Mode registers 1 and 2.
2. If Time-of-Day is to count up, load 0000 in Load registers 1 and 2 and execute command FF43 (Load) to load this value into the counters. This step conditions the count circuitry.
3. Load the desired start time into the Load registers and execute command FF43 again.
4. For counting up, load Load registers 1 and 2 with 0000.
5. Counters 1 and 2 may now be armed.

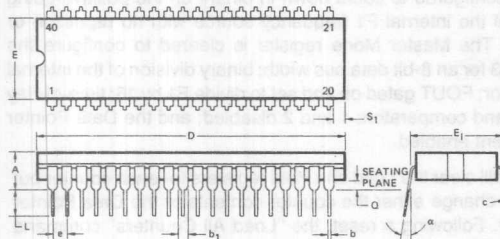


MOS-185

Figure 25. Driving the X1 and X2 Inputs.

PHYSICAL DIMENSIONS

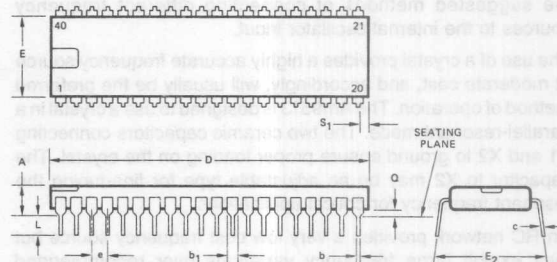
40-Pin Cerdip



Reference Symbol	Inches	
	Min.	Max.
A	.150	.225
b	.016	.020
b ₁	.045	.065
c	.009	.011
D	2.020	2.100
E	.510	.550
E ₁	.600	.630
e	.090	.110
L	.120	.150
Q	.015	.060
S ₁ *	.005	
α	3°	13°

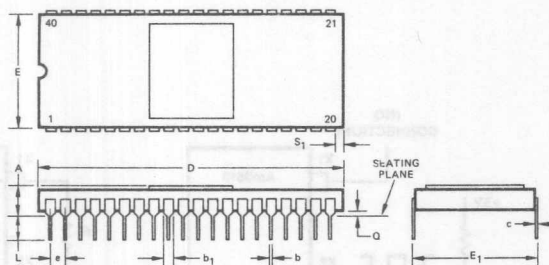
*From edge of end lead.

40-Pin Molded DIP



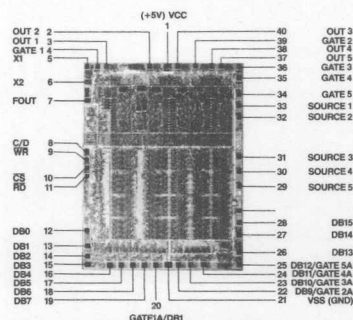
Reference Symbol	Inches	
	Min.	Max.
A	.150	.200
b	.015	.020
b ₁	.055	.065
c	.009	.011
D	2.050	2.080
E	.530	.550
E ₂	.585	.700
e	.090	.110
L	.015	.060
Q	.015	.060
S ₁	.040	.070

40-Pin Side-Brazed Ceramic



Reference Symbol	Inches	
	Min.	Max.
A	.100	.200
b	.015	.022
b ₁	.030	.060
c	.008	.013
D	1.960	2.040
E	.550	.610
E ₁	.590	.620
e	.090	.110
L	.120	.160
Q	.020	.060
S ₁	.005	

Metallization and Pad Layout



DIE SIZE 0.185" x 0.226"

Am9517A

Multimode DMA Controller

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count.
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent autoinitialization of all channels
- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Master system disable
- Enable/disable control of individual DMA requests
- Directly expandable to any number of channels
- End of Process input for terminating transfers
- Software DMA requests
- Independent polarity control for DREQ and DACK signals
- Compressed timing option speeds transfers — up to 2M words/second
- +5 volt power supply
- Advanced N-channel silicon gate MOS technology
- 40 pin Hermetic DIP package
- 100% MIL-STD-883 reliability assurance testing

GENERAL DESCRIPTION

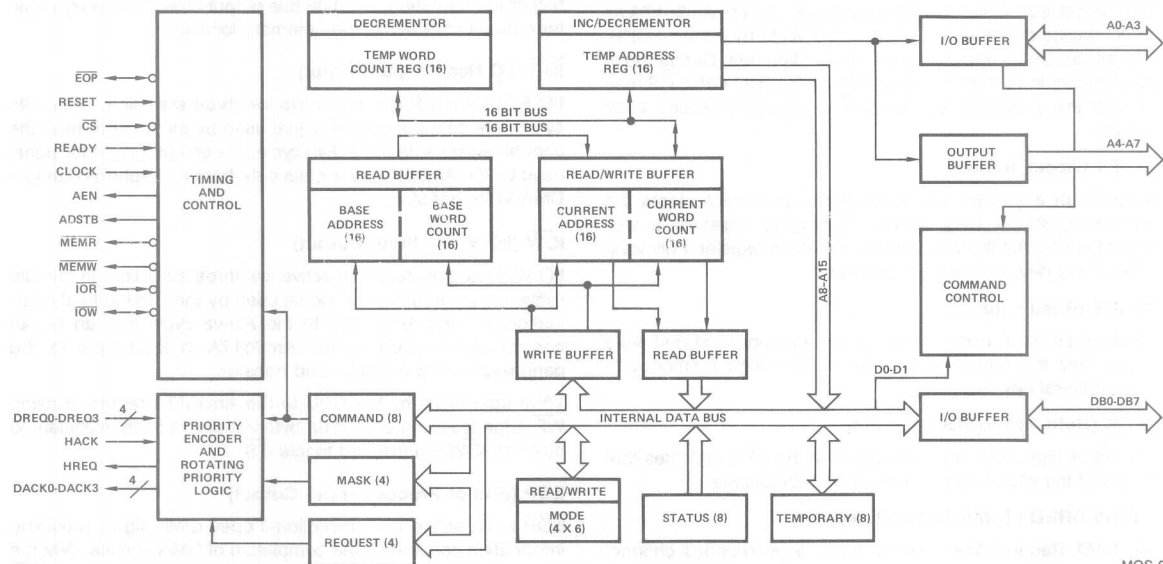
The Am9517A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The Am9517A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The Am9517A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process (EOP).

Each channel has a full 64K address and word count capability. An external EOP signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM

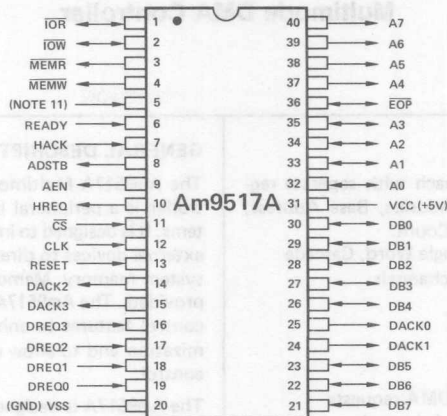


MOS-033

ORDERING INFORMATION

Package Type	Ambient Temperature	Maximum Clock Frequency	
		3MHz	4MHz
Hermetic DIP/ Molded DIP	0°C ≤ T _A ≤ +70°C	AM9517ADC/PC AM9517A-1DC/PC	AM9517A-4DC/PC
Hermetic DIP	-55°C ≤ T _A ≤ +125°C	AM9517ADM	

CONNECTION DIAGRAM



Top View

Pin 1 is marked for orientation.

Figure 1.

MOS-034

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt Supply**VSS:** Ground**CLK (Clock, Input)**

This input controls the internal operations of the Am9517A and its rate of data transfers. The input may be driven at up to 3MHz for the standard Am9517A and up to 4MHz for the Am9517A-4.

CS (Chip Select, Input)

Chip Select is an active low input used to select the Am9517A as an I/O device during an I/O Read or I/O Write by the host CPU. This allows CPU communication on the data bus. During multiple transfers to or from the Am9517A by the host CPU, CS may be held low providing IOR or IOW is toggled following each transfer.

RESET (Reset, Input)

Reset is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.

READY (Ready, Input)

Ready is an input used to extend the memory read and write pulses from the Am9517A to accommodate slow memories or I/O peripheral devices.

HACK (Hold Acknowledge, Input)

The active high Hold Acknowledge from the CPU indicates that control of the system buses has been relinquished.

DREQ0-DREQ3 (DMA Request, Input)

The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. Polarity of DREQ is programmable. Reset initializes these lines to active high.

DB0-DB7 (Data Bus, Input/Output)

The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled during the I/O Read by the host CPU, permitting the CPU to examine

the contents of an Address register, the Status register, the Temporary register or a Word Count register. The Data Bus is enabled to input data during a host CPU I/O write, allowing the CPU to program the Am9517A control registers. During DMA cycles the most significant eight bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations data from the source memory location comes into the Am9517A's Temporary register on the read-from-memory half of the operation. On the write-to-memory half of the operation, the data bus outputs the Temporary register data into the destination memory location.

IOR (I/O Read, Input/Output)

I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the Am9517A to access data from a peripheral during a DMA Write transfer.

IOW (I/O Write, Input/Output)

I/O Write is a bidirectional active low three-state line. In the Idle cycle it is an input control signal used by the CPU to load information into the Am9517A. In the Active cycle it is an output control signal used by the Am9517A to load data to the peripheral during a DMA Read transfer.

Write operations by the CPU to the Am9517A require a rising WR edge following each data byte transfer. It is not sufficient to hold the IOW pin low and toggle CS.

EOP (End of Process, Input/Output)

EOP is an active low bidirectional open-drain signal providing information concerning the completion of DMA service. When a channel's Word Count goes to zero, the Am9517A pulses EOP low to provide the peripheral with a completion signal. EOP may also be pulled low by the peripheral to cause premature completion. The reception of EOP, either internal or external, causes the currently active channel to terminate the service, to set its TC bit in the Status register and to reset its request bit. If Autoinitialization is selected for the channel, the current registers will be updated from the base registers. Otherwise the channel's mask bit will be set and the register contents will remain unaltered.

During memory-to-memory transfers, \overline{EOP} will be output when the TC for channel 1 occurs. \overline{EOP} always applies to the channel with an active DACK; external \overline{EOP} s are disregarded in DACK0-DACK3 are all inactive.

Because \overline{EOP} is an open-drain signal, an external pullup resistor is required. Values of 3.3K or 4.7K are recommended; the \overline{EOP} pin can not sink the current passed by a 1K pullup.

A0-A3 (Address, Input/Output)

The four least significant address lines are bidirectional 3-state signals. During DMA Idle cycles they are inputs and allow the host CPU to load or read control registers. When the DMA is active, they are outputs and provide the lower 4-bits of the output address.

A4-A7 (Address, Output)

The four most significant address lines are three-state outputs and provide four bits of address. These lines are enabled only during DMA service.

HREQ (Hold Request, Output)

The Hold Request to the CPU is used by the DMA to request control of the system bus. Software requests or unmasked DREQs cause the Am9517A to issue HREQ.

DACK0-DACK3 (DMA Acknowledge, Output)

The DMA Acknowledge lines indicate that a channel is active. In many systems they will be used to select a peripheral. Only one DACK will be active at a time and none will be active unless the DMA is in control of the bus. The polarity of these lines is programmable. Reset initializes them to active-low.

AEN (Address Enable, Output)

Address Enable is an active high signal used to disable the system bus during DMA cycles to enable the output of the external latch which holds the upper byte of the address. Note that during DMA transfers HACK and AEN should be used to deselect all other I/O peripherals which may erroneously be accessed as programmed I/O during the DMA operation. The Am9517A automatically deselects itself by disabling the \overline{CS} input during DMA transfers.

ADSTB (Address Strobe, Output)

The active high Address Strobe is used to strobe the upper address byte from DB0-DB7 into an external latch.

MEMR (Memory Read, Output)

The Memory Read signal is an active low three-state output used to access data from the selected memory location during a memory-to-peripheral or a memory-to-memory transfer.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 2. Am9517A Internal Registers.

MEMW (Memory Write, Output)

The Memory Write signal is an active low three-state output used to write data to the selected memory location during a peripheral-to-memory or a memory-to-memory transfer.

FUNCTIONAL DESCRIPTION

The Am9517A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The Am9517A contains 344 bits of internal memory in the form of registers. Figure 2 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

The Am9517A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the Am9517A. The Program Command Control block decodes the various commands given to the Am9517A by the microprocessor prior to servicing a DMA Request. It also decodes each channel's Mode Control word. The Priority Encoder block resolves priority contention among DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In Am9080A systems this input will usually be the $\phi 2$ TTL clock from an Am8224. However, any appropriate system clock will suffice.

DMA Operation

The Am9517A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The Am9517A can assume seven separate states, each composed of one full clock period. State 1 (S1) is the inactive state. It is entered when the Am9517A has no valid DMA requests pending. While in S1, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The Am9517A has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted before S4 by the use of the Ready line on the Am9517A.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for each complete transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23 and S24) for the write-to-memory half of the transfer. The Temporary Data register is used for intermediate storage of the memory byte.

Idle Cycle

When no channel is requesting service, the Am9517A will enter the Idle cycle and perform "S1" states. In this cycle the Am9517A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample \overline{CS} , looking for an attempt by the microprocessor to write or read the internal registers of the Am9517A. When \overline{CS} is low and HACK is low the Am9517A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0-A3 are inputs to the device and select which registers will be read or written. The \overline{IOR} and \overline{IOW} lines are used to select and time reads or writes. Due to the

number and size of the internal registers, an internal flip/flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip/flop is reset by Master Clear or Reset. A separate software command can also reset this flip/flop.

Special software commands can be executed by the Am9517A in the Program Condition. These commands are decoded as sets of addresses when both CS and IOW are active and do not make use of the data bus. Functions include Clear First/Last Flip/Flop and Master Clear.

ACTIVE CYCLE

When the Am9517A is in the Idle cycle and a channel requests a DMA service, the device will output a HREQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode: In Single Transfer mode, the Am9517A will make a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ will go active. After the CPU responds by driving HACK active, a one-byte transfer will take place. Following the transfer, HREQ will go inactive, the word count will be decremented and the address will be either incremented or decremented. When the word count goes to zero a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ will go inactive following each transfer and then will go active again and a new one-byte transfer will be made following each rising edge of HACK. In 8080A/9080A systems this will ensure one full machine cycle of execution between DMA transfers. Details of timing between the Am9517A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode: In Block Transfer mode, the Am9517A will continue making transfers until a TC (caused by the word count going to zero) or an external End of Process (EOP) is encountered. DREQ need be held active only until DACK becomes active. An autoinitialize will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode: In Demand Transfer mode the device will continue making transfers until a TC or external EOP is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count may be read from the Am9517A Current Address and Current Word Count registers. Autoinitialization will only occur following a TC or EOP at the end of service. Following Autoinitialization, an active-going DREQ edge is required to initiate a new DMA service.

Cascade Mode: This mode is used to cascade more than one Am9517A together for simple system expansion. The HREQ and HACK signals from the additional Am9517A are connected to the DREQ and DACK signals of a channel of the initial Am9517A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel in the initial device is used only for prioritizing the additional device, it does not output any address or control

signals of its own. These would conflict with the outputs of the active channel in the added device. The Am9517A will respond to DREQ with DACK but all other outputs except HREQ will be disabled.

Figure 3 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More Am9517As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices forming a third level.

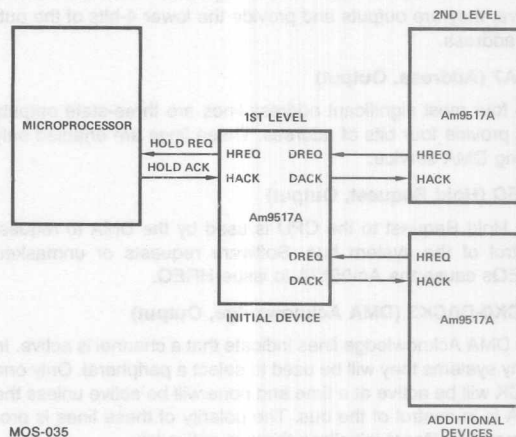


Figure 3. Cascaded Am9517As.

TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating \overline{IOR} and \overline{MEMW} . Read transfers move data from memory to an I/O device by activating \overline{MEMR} and \overline{IOW} . Verify transfers are pseudo transfers; the Am9517A operates as in Read or Write transfers generating addresses, responding to EOP, etc., however, the memory and I/O control lines remain inactive.

Memory-to-Memory: The Am9517A includes a block move capability that allows blocks of data to be moved from one memory address space to another. When Bit C0 in the Command register is set to a logical 1, channels 0 and 1 will operate as memory-to-memory transfer channels. Channel 0 forms the source address and channel 1 forms the destination address. The channel 1 word count is used. A memory-to-memory transfer is initiated by setting a software DMA request for channel 0. Block Transfer Mode should be used for memory-to-memory. When channel 0 is programmed for a fixed source address, a single source word may be written into a block of memory.

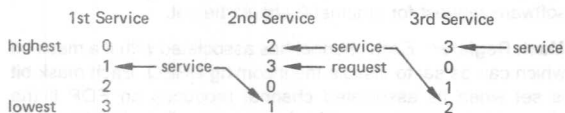
When setting up the Am9517A for memory-to-memory operation, it is suggested that both channels 0 and 1 be masked out. Further, the channel 0 word count should be initialized to the same value used in channel 1. No DACK outputs will be active during memory-to-memory transfers.

The Am9517A will respond to external EOP signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers may be found in Timing Diagram 4.

Autoinitialize: By programming a bit in the Mode register a channel may be set up for an Autoinitialize operation. During Autoinitialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following EOP. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set by EOP when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to repeat its service without CPU intervention.

Priority: The Am9517A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly. With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.



The priority encoder selects the highest priority channel requesting service on each active-going HACK edge. Once a channel is started, its operation will not be suspended if a request is received by a higher priority channel. The high priority channel will only gain control after the lower priority channel releases HREQ. When control is passed from one channel to another, the CPU will always gain bus control. This ensures generation of rising HACK edge to be used to initiate selection of the new highest-priority requesting channel.

Compressed Timing: In order to achieve even greater throughput where system characteristics permit, the Am9517A can compress the transfer time to two clock cycles. From Timing Diagram 3 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3 the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8-A15 need updating (see Address Generation). Timing for compressed transfers is found in Timing Diagram 6.

Address Generation: In order to reduce pin count, the Am9517A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address

bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a 3-state enable. The lower order address bits are output by the Am9517A directly. Lines A0-A7 should be connected to the address bus. Timing Diagram 3 shows the time relationships between CLK, AEN, ADSTB, DB0-DB7 and A0-A7.

During Block and Demand Transfer mode services which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the Am9517A executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services, S \star states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

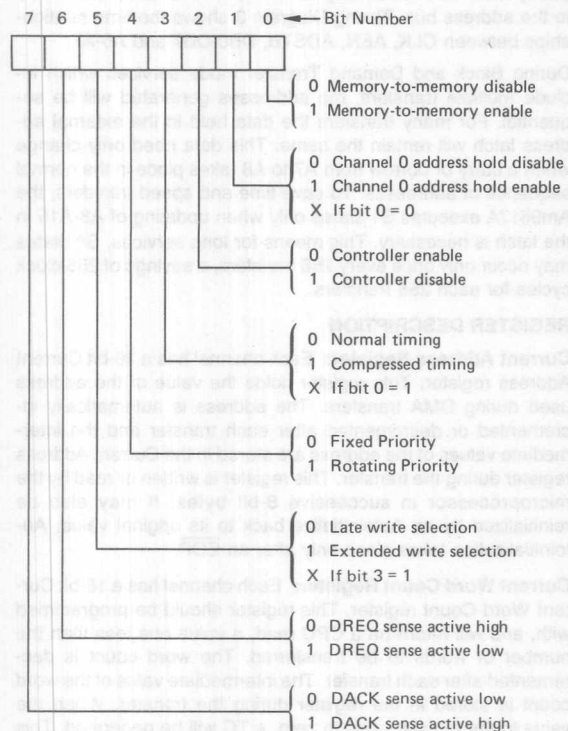
REGISTER DESCRIPTION

Current Address Register: Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an EOP.

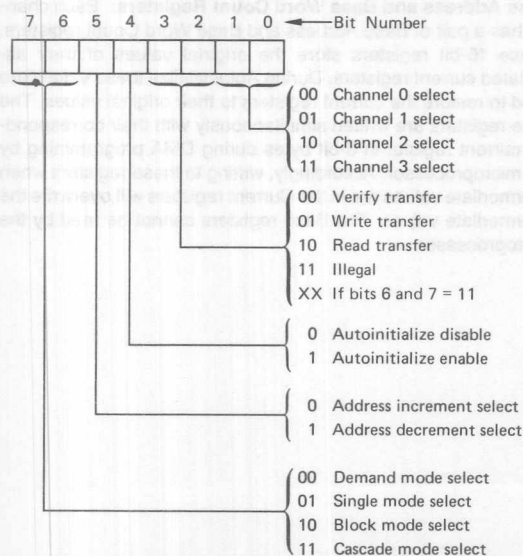
Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register should be programmed with, and will return on a CPU read, a value one less than the number of words to be transferred. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an EOP occurs. Note that the contents of the Word Count register will be FFFF (hex) following on internally generated EOP.

Base Address and Base Word Count Registers: Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes during DMA programming by the microprocessor. Accordingly, writing to these registers when intermediate values are in the Current registers will overwrite the intermediate values. The Base registers cannot be read by the microprocessor.

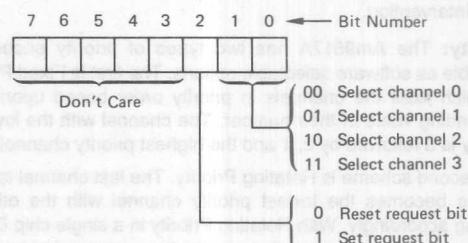
Command Register: This 8-bit register controls the operation of the Am9517A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 4 for address coding.



Mode Register: Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register it to be written.

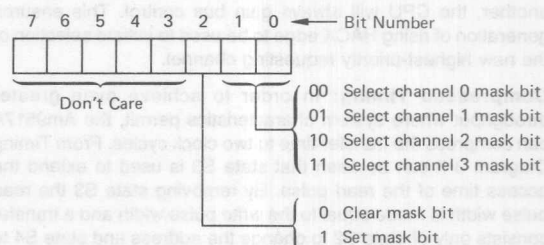


Request Register: The Am9517A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 4 for address coding.

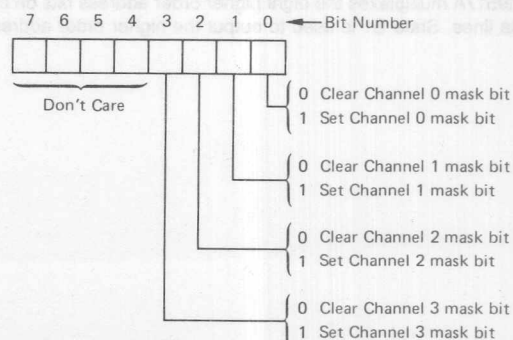


Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

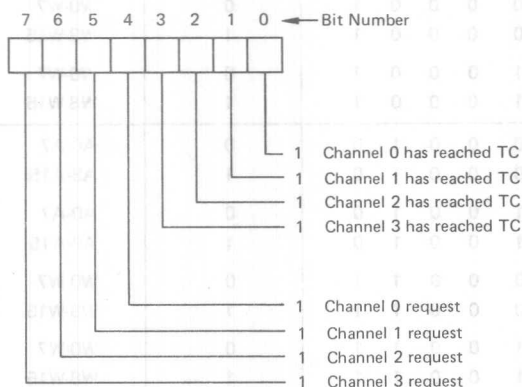
Mask Register: Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 4 for instruction addressing.



All four bits of the Mask Register may also be written with a single command.



Status Register: The Status registers may be read out of the Am9517A by the microprocessor. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.



Temporary Register: The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands: There are two special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The two software commands are:

Clear First/Last Flip/Flop: This command may be issued prior to writing or reading Am9517A address or word count information. This initializes the flip/flop to a known state so that subsequent accesses to register contents by the microprocessor will address lower and upper bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The Am9517A will enter the Idle cycle.

Figure 4 lists the address codes for the software commands.

Interface Signals						Operation
A3	A2	A1	A0	IOR	IOW	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Illegal
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 4. Register and Function Addressing.

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0—DB7
			CS	IOR	IOW	A3	A2	A1	A0		
0	Base & Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base & Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base & Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base & Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

Figure 5. Word Count and Address Register Command Codes.

MAXIMUM RATINGS above which useful life may be impaired

Storage Temperature	−65°C to +150°C
Ambient Temperature Under Bias	−55°C to +125°C
VCC with Respect to VSS	−0.5V to +7.0V
All Signal Voltages with Respect to VSS	−0.5V to +7.0V
Power Dissipation (Package Limitation)	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	T _A	VCC
Am9517ADC/PC	0°C to +70°C	5.0V ±5%
Am9517A-1DC/PC	0°C to +70°C	5.0V ±5%
Am9517A-4DC/PC	0°C to +70°C	5.0V ±5%
Am9517ADM	−55°C to +125°C	5.0V ±10%

ELECTRICAL CHARACTERISTICS over operating range (Note 1)

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
VOH	Output HIGH Voltage	I _{OH} = −200μA	2.4			Volts
		I _{OH} = −100μA, (HREQ Only)	3.3			
VOL	Output LOW Voltage	I _{OL} = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC+0.5	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC	−10		+10	μA
IOZ	Output Leakage Current	VCC ≤ VO ≤ VSS+40	−10		+10	μA
ICC	VCC Supply Current	T _A = +25°C		65	130	mA
		T _A = 0°C		75	150	
		T _A = −55°C			175	
CO	Output Capacitance	f _c = 1.0MHz, Inputs = 0V		4	8	pF
CI	Input Capacitance			8	15	pF
CIO	I/O Capacitance			10	18	pF

NOTES:

- Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
- Input timing parameters assume transition times of 20ns or less. Waveform measurement points for both input and output signals are 2.0V for High and 0.8V for Low, unless otherwise noted.
- Output loading is 1 Standard TTL gate plus 50pF capacitance unless noted otherwise.
- The new IOW or MEMW pulse width for normal write will be TCY-100ns and for extended write will be 2TCY-100ns. The net IOR or MEMR pulse width for normal read will be 2TCY-50ns and for compressed read will be TCY-50ns.
- TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3kΩ pull-up resistor connected from HREQ to VCC.
- DREQ should be held active until DACK is returned.
- DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
- Output loading on the data bus is 1 Standard TTL gate plus 15pF for the minimum value and 1 Standard TTL gate plus 100pF for the maximum value.
- Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600ns for the Am9517A or Am9517A-1 and at least 450ns for the Am9517A-4 as recovery time between active read or write pulses.
- Parameters are listed in alphabetical order.
- Pin 5 is an input that should always be at a logic high level. An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to VCC.
- Signals $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ refer to $\overline{\text{IOR}}$ and $\overline{\text{MEMW}}$ respectively for peripheral-to-memory DMA operations and to $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ respectively for memory-to-peripheral DMA operations.
- If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).

Am9517A

SWITCHING CHARACTERISTICS

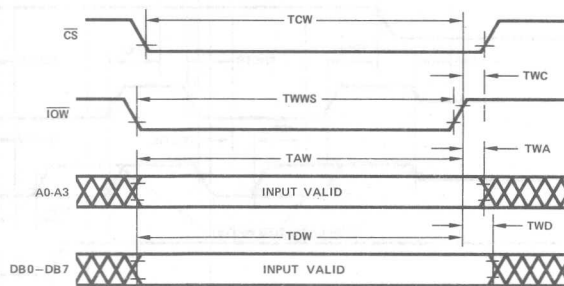
ACTIVE CYCLE (Notes 2, 3, 10, 11 and 12)

Parameter	Description	Am9517A		Am9517A-1		Am9517A-4		Unit
		Min	Max	Min	Max	Min	Max	
TAEI	AEN HIGH from CLK LOW (S1) Delay Time		300		300		225	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		200		200		150	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		150		120	ns
TAFC	READ or WRITE Float from CLK HIGH		150		150		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		250		190	ns
TAHR	ADR from READ HIGH Hold Time	TCY-100		TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	50		50		40		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		280		280		220	ns
	EOP HIGH from CLK HIGH Delay Time		250		250		190	ns
	EOP LOW to CLK HIGH Delay Time		250		250		190	ns
TASM	ADR Stable from CLK HIGH		250		250		190	ns
TASS	DB to ADSTB LOW Setup Time	100		100		100		ns
TCH	Clock High Time (Transitions ≤ 10 ns)	120		120		100		ns
TCL	Clock Low Time (Transitions ≤ 10 ns)	150		150		110		ns
TCY	CLK Cycle Time	320		320		250		ns
TDCL	CLK HIGH to READ or WRITE LOW Delay (Note 4)		270		270		200	ns
TDCTR	READ HIGH from CLK HIGH (S4) Delay Time (Note 4)		270		270		210	ns
TDCTW	WRITE HIGH from CLK HIGH (S4) Delay Time (Note 4)		200		200		150	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 5)		160		160		120	ns
TDQ2			250		250		190	ns
TEPS	EOP LOW from CLK LOW Setup Time	60		60		45		ns
TEPW	EOP Pulse Width	300		300		225		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		250		190	ns
TFAC	READ or WRITE Active from CLK HIGH		200		200		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		300		225	ns
THS	HACK valid to CLK HIGH Setup Time	100		100		75		ns
TIDH	Input Data from MEMR HIGH Hold Time	0		0		0		ns
TIDS	Input Data to MEMR HIGH Setup Time	250		250		190		ns
TODH	Output Data from MEMW HIGH Hold Time	20		20		20		ns
TODV	Output Data Valid to MEMW HIGH (Note 13)	200		200		125		ns
TQS	DREQ to CLK LOW (S1, S4) Setup Time	120		120		90		ns
TRH	CLK to READY LOW Hold Time	20		20		20		ns
TRS	READY to CLK LOW Setup Time	100		100		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		200		150	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		140		110	ns

SWITCHING CHARACTERISTICS (Cont.)**PROGRAM CONDITION (IDLE CYCLE)**

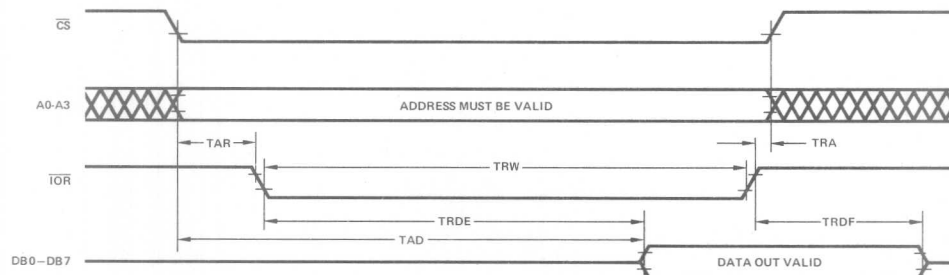
(Notes 2, 3, 10, 11 and 12)

Parameter	Description	Am9517A		Am9517A-1		Am9517A-4		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
TAR	ADR Valid or \overline{CS} LOW to \overline{READ} LOW	50		50		50		ns
TAW	ADR Valid to \overline{WRITE} HIGH Setup Time	200		200		150		ns
TCW	\overline{CS} LOW to \overline{WRITE} HIGH Setup Time	200		200		150		ns
TDW	Data Valid to \overline{WRITE} HIGH Setup Time	200		200		150		ns
TRA	ADR or \overline{CS} Hold from \overline{READ} HIGH	0		0		0		ns
TRDE	Data Access from \overline{READ} LOW (Note 8)		300		200		200	ns
TDRF	DB Float Delay from \overline{READ} HIGH	20	150	20	100	20	100	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		500		μ s
TRSTS	RESET to First \overline{IOWR}	2		2		2		TCY
TRSTW	RESET Pulse Width	300		300		300		ns
TRW	\overline{READ} Width	300		300		250		ns
TWA	ADR from \overline{WRITE} HIGH Hold Time	20		20		20		ns
TWC	\overline{CS} HIGH from \overline{WRITE} HIGH Hold Time	20		20		20		ns
TWD	Data from \overline{WRITE} HIGH Hold Time	30		30		30		ns
TWWS	Write Width	200		200		200		ns
TAD	Data Access from ADR Valid, \overline{CS} LOW		350		300		300	ns

SWITCHING WAVEFORMS

Timing Diagram 1. Program Condition Write Timing (Note 9).

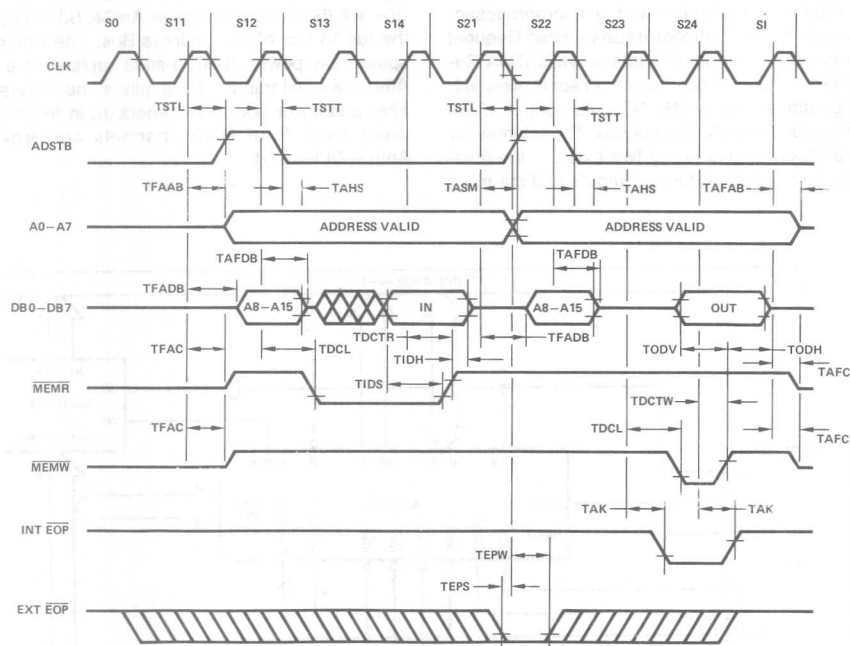
MOS-036



Timing Diagram 2. Program Condition Read Cycle (Note 9).

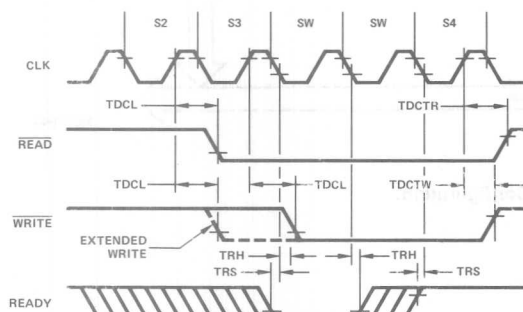
MOS-037

SWITCHING WAVEFORMS (Cont.)



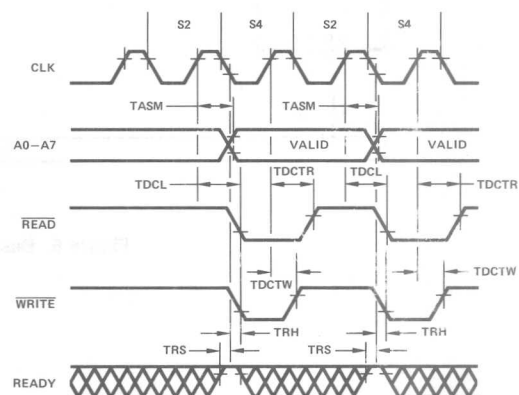
Timing Diagram 4. Memory-to-Memory.

MOS-039



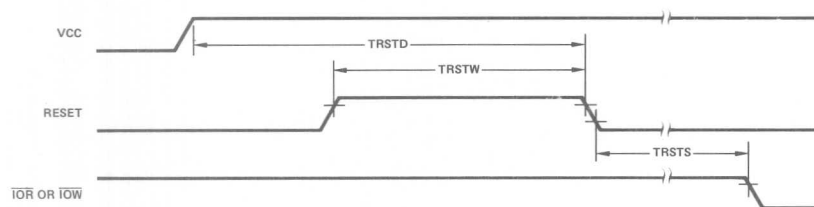
Timing Diagram 5. Ready Timing.

MOS-040



Timing Diagram 6. Compressed Timing.

MOS-041



Timing Diagram 7. Reset Timing.

MOS-042

APPLICATION INFORMATION

Figure 6 shows a convenient method for configuring a DMA system with the Am9517A Controller and a microprocessor system. The Multimode DMA Controller issues a Hold Request to the processor whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A takes control of the Address Bus, the Data Bus and the Control Bus. The address for the first transfer operation comes out in two bytes – the least significant eight bits on the eight Address outputs and the most

significant eight bits on the Data Bus. The contents of the Data Bus are then latched into the Am74LS373 register to complete the full 16 bits of the Address Bus. The Am74LS373 is a high speed, low power, 8-bit, 3-state register in a 20-pin package. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one Am9517A is used.

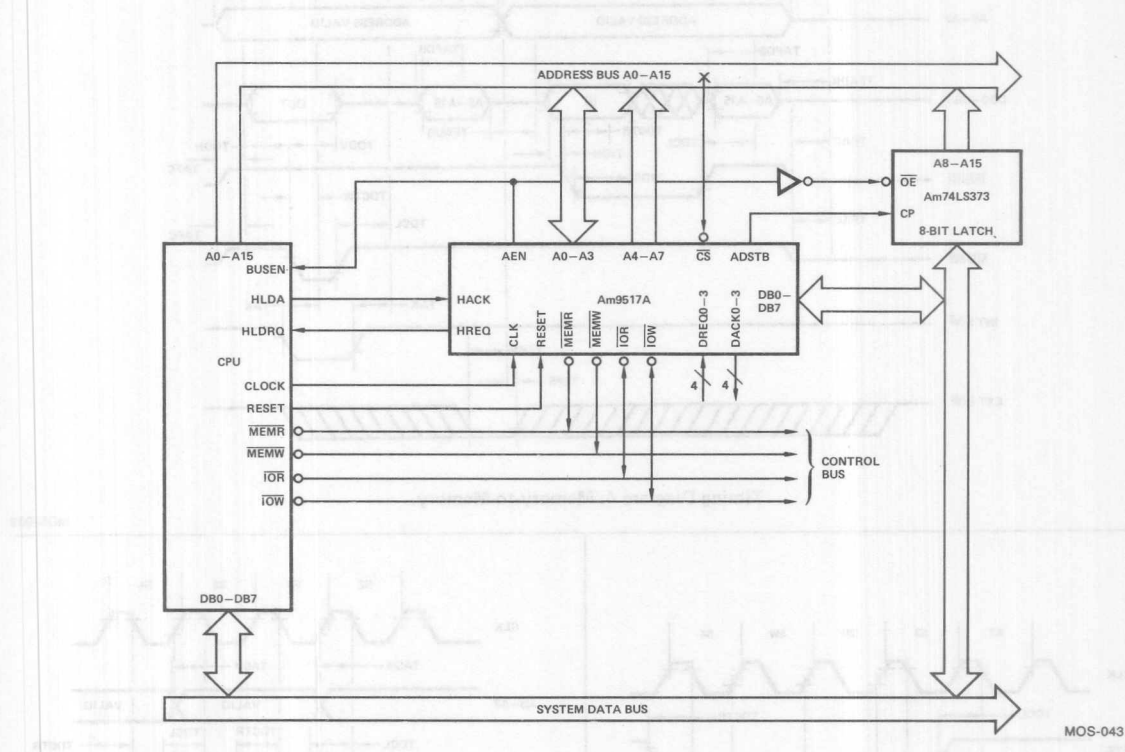
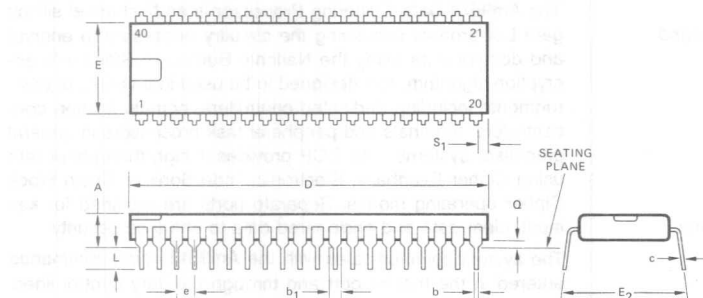


Figure 6. Basic DMA Configuration.

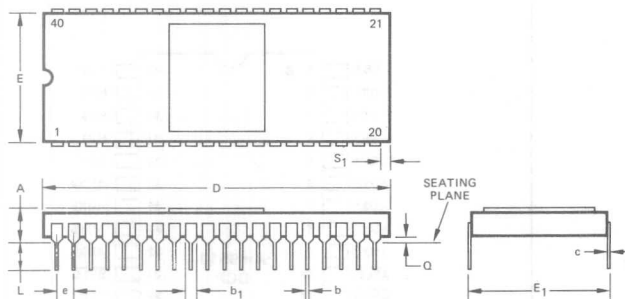
PHYSICAL DIMENSIONS Dual-In-Line

40-Pin Plastic



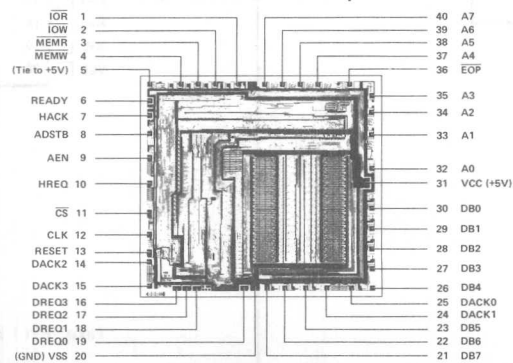
Reference Symbol	Inches	
	Min.	Max.
A	.150	.200
b	.015	.020
b ₁	.055	.065
c	.009	.011
D	2.050	2.080
E	.530	.550
E ₁	.585	.700
e	.090	.110
L	.125	.160
Q	.015	.060
S ₁	.040	.070

40-Pin Hermetic



Reference Symbol	Inches	
	Min.	Max.
A	.100	.200
b	.015	.022
b ₁	.030	.060
c	.008	.013
D	1.960	2.040
E	.550	.610
E ₁	.590	.620
e	.090	.110
L	.120	.160
Q	.020	.060
S ₁	.005	

Metallization and Pad Layout



DIE SIZE
0.198" X 0.210"

Am9518

Data Ciphering Processor

ADVANCED DATA

DISTINCTIVE CHARACTERISTICS

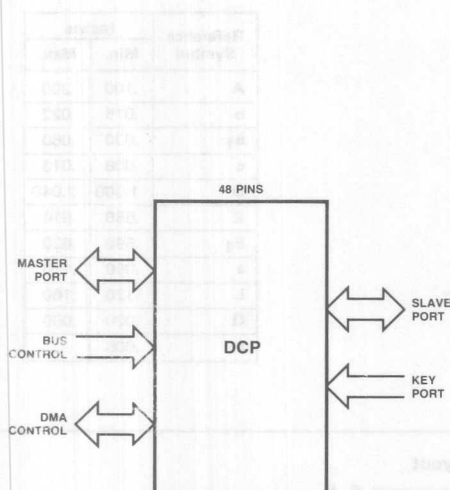
- Standard encryption and decryption algorithms
- Throughput rates greater than 1 megabyte per second
- Supports three standard ciphering options
 - Electronic Code Book
 - Cipher Feedback
 - Chain Block
- Master, Encrypt and Decrypt key registers
- Key parity check
- Separate key port
- Session keys and Initializing Vectors may be entered encrypted or clear
- Master data port for bidirectional bus operation
- Slave data port for pipelined operation
- 40-pin DIP package
- +5V power supply

GENERAL DESCRIPTION

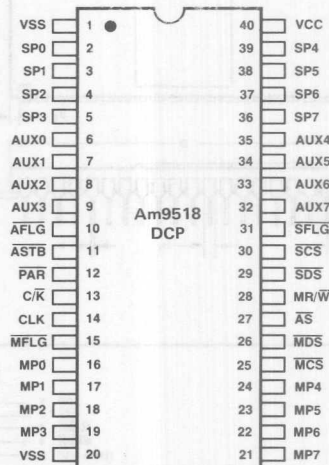
The Am9518 Data Ciphering Processor is an N-channel silicon gate LSI product containing the circuitry necessary to encrypt and decrypt data using the National Bureau of Standards encryption algorithm. It is designed to be used in a variety of environments including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems. The DCP provides a high throughput rate using Cipher Feedback, Electronic Code Book or Chain Block Cipher operating modes. Separate ports are provided for key input, clear data and enciphered data to enhance security.

The system communicates with the Am9518 using commands entered in the master port and through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities are all performed concurrently. External DMA control can easily be used to enhance throughput in some system configurations. This device is designed to interface directly to the AmZ8000 CPU bus and, with a minimum of external logic, to the 2900, 8080, 8085 and 8048 families of processors.

INTERFACE FLOW

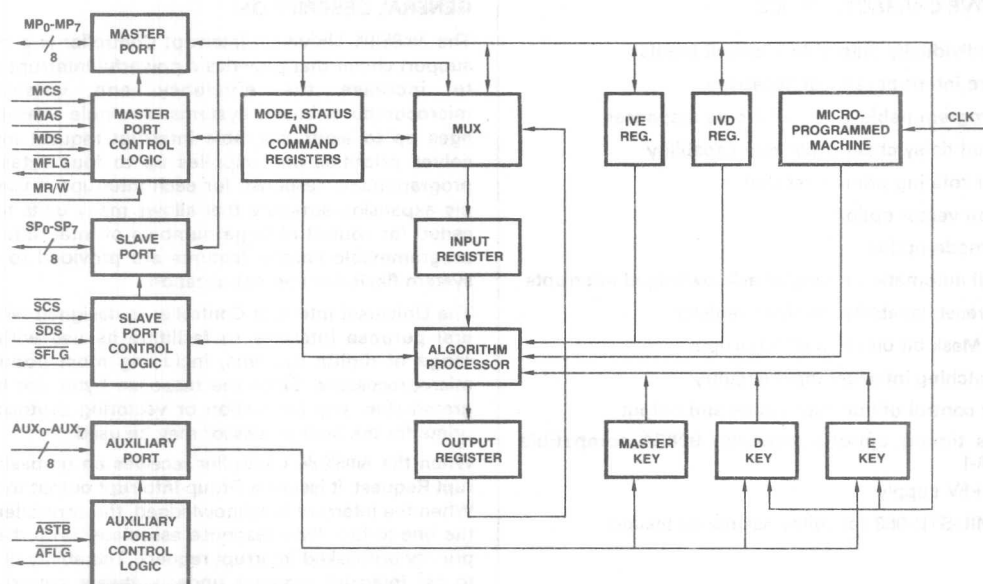


CONNECTION DIAGRAM Top View



Note: Pin 1 is marked for orientation.

BLOCK DIAGRAM



Am9519A

Universal Interrupt Controller

DISTINCTIVE CHARACTERISTICS

- Eight individually maskable interrupt inputs
- Software interrupt request capability
- Fully programmable 1, 2, 3 or 4 byte responses
- Unlimited daisy-chain expansion capability
- Fixed or rotating priority resolution
- Common vector option
- Polled mode option
- Optional automatic clearing of acknowledged interrupts
- Bit set/reset capability for Mask register
- Master Mask bit disables all interrupts
- Pulse-catching interrupt input circuitry
- Polarity control of interrupt inputs and output
- Various timing options including 8085A compatible Am9519A-1
- Single +5V supply
- 100% MIL-STD-883 reliability assurance testing

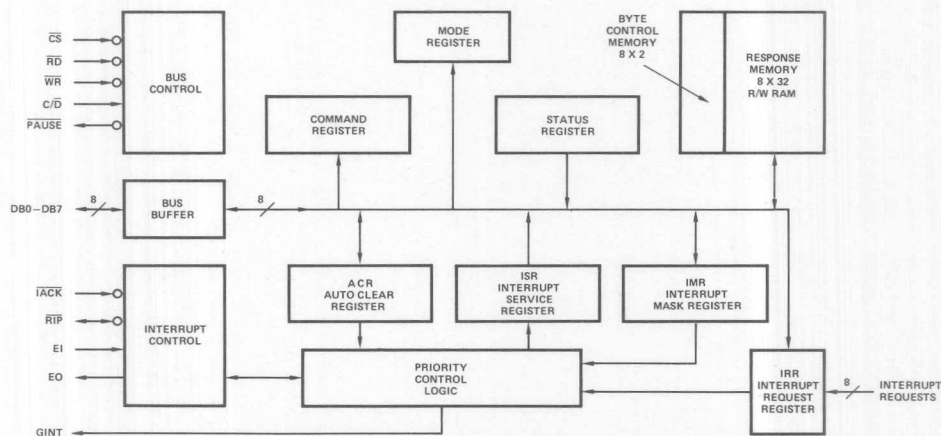
GENERAL DESCRIPTION

The Am9519A Universal Interrupt Controller is a processor support circuit that provides a powerful interrupt structure to increase the efficiency and versatility of microcomputer-based systems. A single Am9519A manages up to eight maskable interrupt request inputs, resolves priorities and supplies up to four bytes of fully programmable response for each interrupt. It uses a simple expansion structure that allows many units to be cascaded for control of large numbers of interrupts. Several programmable control features are provided to enhance system flexibility and optimization.

The Universal Interrupt Controller is designed with a general purpose interface to facilitate its use with a wide range of digital systems, including most popular 8-bit microprocessors. Since the response bytes are fully programmable, any instruction or vectoring protocol appropriate for the host processor may be used.

When the Am9519A controller receives an unmasked Interrupt Request, it issues a Group Interrupt output to the CPU. When the interrupt is acknowledged, the controller outputs the one-to-four byte response associated with the highest priority unmasked interrupt request. The ability of the CPU to set interrupt requests under software control permits hardware prioritization of software tasks and aids system diagnostic and maintenance procedures.

BLOCK DIAGRAM



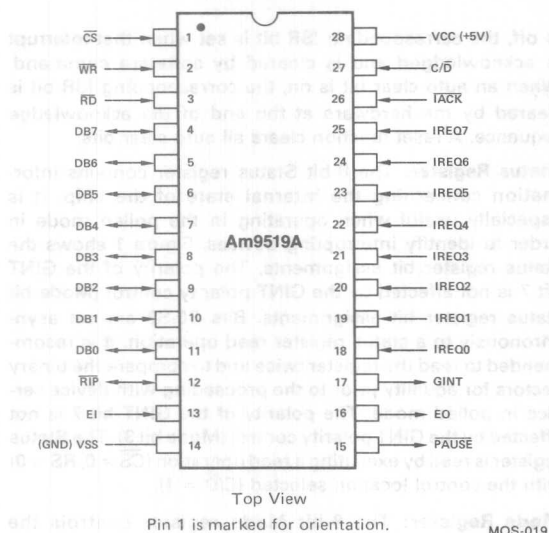
MOS-143

ORDERING INFORMATION

Package Type	Ambient Temperature	Timing Options	
		Am9519A	Am9519A-1
Hermetic DIP*	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	AM9519ADC/CC	AM9519A-1DC/CC
	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	AM9519ADM	
Molded DIP	$0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$	AM9519APC	AM9519A-1PC

*DC = Side-Braced Ceramic CC = Cerdip

CONNECTION DIAGRAM



INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt Power Supply

VSS: Ground

DB0 – DB7 (Data Bus, Input/Output)

The eight bidirectional data bus signals are used to transfer information between the Am9519A and the system data bus. The direction of transfer is controlled by the $\overline{\text{IACK}}$, $\overline{\text{WR}}$ and $\overline{\text{RD}}$ input signals. Programming and control information are written into the device; status and response data are output by it.

 $\overline{\text{CS}}$ (Chip Select, Input)

The active low Chip Select input enables read and write operations on the data bus. Interrupt acknowledge responses are not conditioned by $\overline{\text{CS}}$.

 $\overline{\text{RD}}$ (Read, Input)

The active low Read signal is conditioned by $\overline{\text{CS}}$ and indicates that information is to be transferred from the Am9519A to the data bus.

 $\overline{\text{WR}}$ (Write, Input)

The active low Write signal is conditioned by $\overline{\text{CS}}$ and indicates that data bus information is to be transferred from the data bus to a location within the Am9519A.

 $\text{C}/\overline{\text{D}}$ (Control/Data, Input)

The $\text{C}/\overline{\text{D}}$ control signal selects source and destination locations for data bus read and write operations. Data read or write transfers are made to or from preselected internal registers or memory locations. Control write operations load the command register and control read operations output the status register.

IREQ0 – IREQ7 (Interrupt Request, Input)

The Interrupt Request signals are used by external devices to indicate that service by the host CPU is desired. IREQ inputs are accepted asynchronously and they may be programmed for either a high-to-low or low-to-high

edge transition. Active inputs are latched internally in the Interrupt Request Register. After the IRR bit is cleared, an IREQ transition of the programmed polarity must occur to initiate another request.

RIP (Response In Process, Input/Output)

Response In Process is a bidirectional signal used when two or more Am9519A circuits are cascaded. It permits multibyte response transfers to be completed without interference from higher priority interrupts. An Am9519A that is responding to an acknowledged interrupt will treat $\overline{\text{RIP}}$ as an output and hold it low until the acknowledge response is finished. An Am9519A without an acknowledged interrupt will treat $\overline{\text{RIP}}$ as an input and will ignore $\overline{\text{IACK}}$ pulses as long as $\overline{\text{RIP}}$ is low. The $\overline{\text{RIP}}$ output is open drain and requires an external pullup resistor to VCC.

 $\overline{\text{IACK}}$ (Interrupt Acknowledge, Input)

The active low Interrupt Acknowledge line indicates that the external system is asking for interrupt response information. Depending on the programmed state of the Am9519A, it will accept 1, 2, 3 or 4 $\overline{\text{IACK}}$ pulses; one response byte is transferred per pulse. The first $\overline{\text{IACK}}$ pulse causes selection of the highest priority unmasked pending interrupt request and generates a $\overline{\text{RIP}}$ output signal.

 $\overline{\text{PAUSE}}$ (Pause, Output)

The active-low Pause signal is used to coordinate interrupt responses with data bus and control timing. Pause goes low when the first $\overline{\text{IACK}}$ is received and remains low until $\overline{\text{RIP}}$ goes low. The external system can use Pause to stretch the acknowledge cycle and allow the control timing to automatically adjust to the actual priority resolution delays in the interrupt system. Second, third and fourth response bytes do not cause Pause to go low. Pause is an open drain output and requires an external pullup resistor to VCC.

EO (Enable Out, Output)

The active high EO signal is used to implement daisy-chained cascading of several Am9519A circuits. EO is connected to the EI input of the next lower priority chip. On receipt of an interrupt acknowledge, each EO will go inactive until it has been determined that no valid interrupt request is pending on that chip. If an active request is present, EO remains low. EO is also held low when the master mask bit is active, thus disabling all lower priority chips.

EI (Enable In, Input)

The active high EI signal is used to implement daisy-chained cascading of several Am9519A circuits. EI is connected to EO of the next higher priority chip. It may also be used as a hardware disable input for the interrupt system. When EI is low $\overline{\text{IACK}}$ inputs are ignored. EI is internally pulled up to VCC so that no external pullup is needed when EI is not used.

GINT (Group Interrupt, Output)

The Group Interrupt output signal indicates that at least one unmasked interrupt request is pending. It may be programmed for active high or active low polarity. When active low, the output is open drain and requires an external pull up resistor to VCC.

REGISTER DESCRIPTION

Interrupt Request Register (IRR): The 8-bit IRR is used to store pending interrupt requests. A bit in the IRR is set whenever the corresponding IREQ input goes active. Bits may also be set under program control from the CPU, thus permitting software generated interrupts. IRR bits may be cleared under program control. An IRR bit is automatically cleared when its interrupt is acknowledged. All IRR bits are cleared by a reset function.

Interrupt Service Register (ISR): The 8-bit ISR contains one bit for each IREQ input. It is used to indicate that a pending interrupt has been acknowledged and to mask all lower priority interrupts. When a bit is set by the acknowledge logic in the ISR, the corresponding IRR bit is cleared. If an acknowledged interrupt is not programmed to be automatically cleared, its ISR bit must be cleared by the CPU under program control when it is desired to permit interrupts from lower priority devices. When the interrupt is programmed for automatic clearing, the ISR bit is automatically reset during the acknowledge sequence. All ISR bits are cleared by a reset function.

Interrupt Mask Register (IMR): The 8-bit IMR is used to enable or disable the individual interrupt inputs. The IMR bits correspond to the IREQ inputs and all eight may be loaded, set or cleared in parallel under program control. In addition, individual IMR bits may be set or cleared by the CPU. A reset function will set all eight mask bits, disabling all requests. A mask bit that is set does not disable the IRR, and an IREQ that arrives while a corresponding mask bit is set will cause an interrupt later when the mask bit is cleared. Only unmasked interrupt inputs can generate a Group Interrupt output.

Response Memory: An 8 x 32 read/write response memory is included in the Am9519A. It is used to store up to four bytes of response information for each of the eight interrupt request inputs. All bits in the memory are programmable, allowing any desired vector, opcode, instruction or other data to be entered. The Am9519A transfers the interrupt response information for the highest priority unmasked interrupt from the memory to the data bus when the IACK input is active.

Auto Clear Register: The 8-bit Auto Clear register contains one bit for each IREQ input and specifies the operating mode for each of the ISR bits. When an auto clear bit

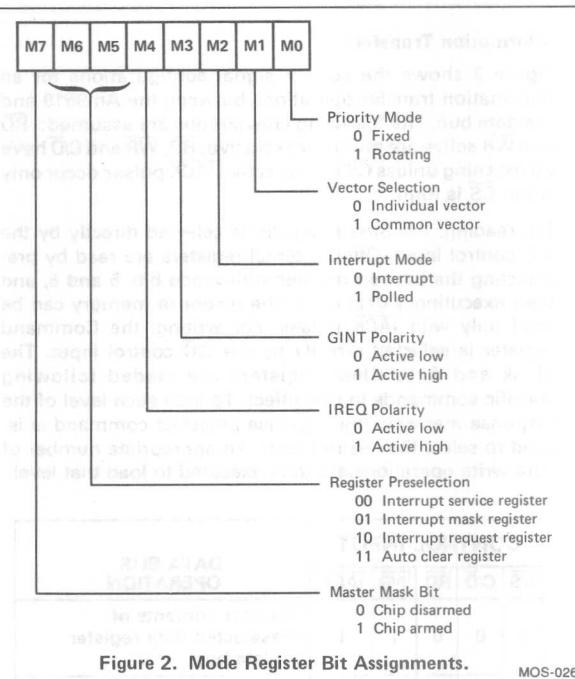
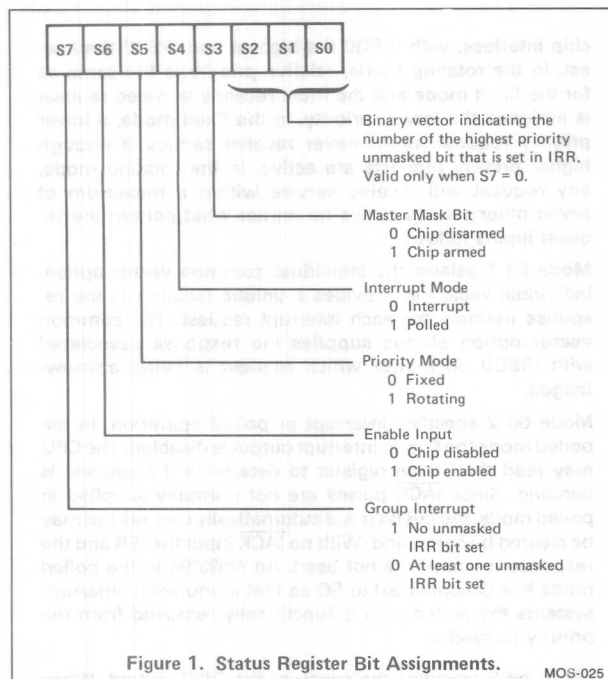
is off, the corresponding ISR bit is set when that interrupt is acknowledged and is cleared by software command. When an auto clear bit is on, the corresponding ISR bit is cleared by the hardware at the end of the acknowledge sequence. A reset function clears all auto clear bits.

Status Register: The 8-bit Status register contains information concerning the internal state of the chip. It is especially useful when operating in the polled mode in order to identify interrupting devices. Figure 1 shows the status register bit assignments. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit status register bit assignments. Bits S0-S2 are set asynchronously to a status register read operation. It is recommended to read the register twice and to compare the binary vectors for equality prior to the proceeding with device service in polled mode. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit 3). The Status register is read by executing a read operation ($\overline{CS} = 0, RS = 0$) with the control location selected ($C/D = 1$).

Mode Register: The 8-bit Mode register controls the operating options of the Am9519A. Figure 2 shows the bit assignments for the Mode register. The five low order mode bits (0 through 4) are loaded in parallel by command. Bits 5, 6 and 7 are controlled by separate commands. (See Figure 4.) The Mode register cannot be read out directly to the data bus, but Mode bits 0, 2 and 7 are available as part of the Status register.

Command Register: The 8-bit Command register stores the last command entered. Depending upon the command opcode, it may initiate internal actions or precondition the part for subsequent data bus transfers. The Command register is loaded by executing a write operation ($\overline{WR} = 0$) with the control location selected ($C/\overline{D} = 1$), as shown in Figure 3.

Byte Count Register: The length in bytes of the response associated with each interrupt is independently programmed so that different interrupts may have different length responses. The byte count for each response is stored in eight 2-bit Byte Count registers. For a given interrupt the Am9519A will expect to receive a number of IACK pulses that equals the corresponding byte count, and will hold RIP low until the count is satisfied.



FUNCTIONAL DESCRIPTION

Interrupts are used to improve system throughput and response time by eliminating heavy dependence on software polling procedures. Interrupts allow external devices to asynchronously modify the instruction sequence of a program being executed. In systems with multiple interrupts, vectoring can further improve performance by allowing direct identification of the interrupting device and its associated service routine. The Am9519A Universal Interrupt Controller contains, on one chip, all of the circuitry necessary to detect, prioritize and manage eight vectored interrupts. It includes many options and operating modes that permit the design of sophisticated interrupt systems.

Reset

The reset function is accomplished by software command or automatically during power-up. The reset command may be issued by the CPU at any time. Internal power up circuitry is triggered when VCC reaches a predetermined threshold, causing a brief internal reset pulse. In both cases, the resulting internal state of the machine is that all registers are cleared except the Mask register which is set. Thus no Group Interrupt will be generated and no interrupt requests will be recognized. The response memory and Byte Count registers are not affected by reset. Their contents after power-up are unpredictable and must be established by the host CPU during initialization.

Operating Sequence

A brief description of a typical sequence of events in an operating interrupt system will illustrate the general interactions among the host CPU, the interrupt controller and the interrupting peripheral.

1. The Am9519A controller is initialized by the CPU in order to customize its configuration and operation for the application at hand. Both the controller and the CPU are then enabled to accept interrupts.

2. One (or more) of the interrupt request inputs to the controller becomes active indicating that peripheral equipment is asking for service. The controller asynchronously accepts and latches the request(s).
3. If the request is masked, no further action takes place. If the request is not masked, a Group Interrupt output is generated by the controller.
4. The GINT signal is recognized by the CPU which normally will complete the execution of the current instruction, insert an interrupt acknowledge sequence into its instruction execution stream, and disable its internal interrupt structure. The controller expects to receive one or more $\overline{\text{IACK}}$ signals from the CPU during the acknowledge sequence.
5. When the controller receives the $\overline{\text{IACK}}$ signal, it brings $\overline{\text{PAUSE}}$ low and selects the highest priority unmasked pending request. When selection is complete, the $\overline{\text{RIP}}$ output is brought low and the first byte in the response memory associated with the selected request is output on the data bus. $\overline{\text{PAUSE}}$ stays low until $\overline{\text{RIP}}$ goes low. $\overline{\text{RIP}}$ stays low until the last byte of the response has been transferred.
6. During the acknowledge sequence, the IRR bit corresponding to the selected request is automatically cleared, and the corresponding ISR bit is set. When the ISR bit is set, the Group Interrupt output is disabled until a higher priority request arrives or the ISR bit is cleared. The ISR bit will be cleared by either hardware or software.
7. If a higher priority request arrives while the current request is being serviced, GINT will be output by the controller, but will be recognized and acknowledged only if the CPU has its interrupt input enabled. If acknowledged, the corresponding higher priority ISR bit will be set and the requests nested.

Information Transfers

Figure 3 shows the control signal configurations for all information transfer operations between the Am9519 and the data bus. The following conventions are assumed: \overline{RD} and \overline{WR} active are mutually exclusive; \overline{RD} , \overline{WR} and $\overline{C/D}$ have no meaning unless \overline{CS} is low; active \overline{IACK} pulses occur only when \overline{CS} is high.

For reading, the Status register is selected directly by the $\overline{C/D}$ control input. Other internal registers are read by preselecting the desired register with mode bits 5 and 6, and then executing a data read. The response memory can be read only with \overline{IACK} pulses. For writing, the Command register is selected directly by the $\overline{C/D}$ control input. The Mask and Auto Clear registers are loaded following specific commands to that effect. To load each level of the response memory, the response preselect command is issued to select the desired level. An appropriate number of data write operations are then executed to load that level.

CONTROL INPUT					DATA BUS OPERATION
\overline{CS}	$\overline{C/D}$	\overline{RD}	\overline{WR}	\overline{IACK}	
0	0	0	1	1	Transfer contents of preselected data register to data bus
0	0	1	0	1	Transfer contents of data bus to preselected data register
0	1	0	1	1	Transfer contents of status register to data bus
0	1	1	0	1	Transfer contents of data bus to command register
1	X	X	X	0	Transfer contents of selected response memory location to data bus
1	X	X	X	1	No information transferred

Figure 3. Summary of Data Bus Transfers.

The Pause output may be used by the host CPU to ensure that proper timing relationships are maintained with the Am9519A when \overline{IACK} is active. The \overline{IACK} pulse width required depends on several variables, including: operating temperature, internal logic delays, number of interrupt controllers chained together, and the priority level of the interrupt being acknowledged. When delays in these variables combine to delay selection of a request following the falling edge of the first \overline{IACK} , the Pause output may be used to extend the \overline{IACK} pulse, if necessary. Pause will remain low until a request has been selected, as indicated by the falling edge of \overline{RIP} . Typically, the internal interrupt selection process is quite fast, especially for systems with a single Am9519A and Pause will consequently remain low for only a very brief interval and will not cause extension of the \overline{IACK} timing.

Operating Options

The Mode register specifies the various combinations of operating options that may be selected by the CPU. It is cleared by power-up or by a reset command.

Mode bit 0 specifies the rotating/fixed priority mode (see Figure 2). In the fixed mode, priority is assigned to the request inputs based upon their physical location at the

chip interface, with $\overline{IREQ0}$ the highest and $\overline{IREQ7}$ the lowest. In the rotating mode, relative priority is the same as for the fixed mode and the most recently serviced request is assigned the lowest priority. In the fixed mode, a lower priority request might never receive service if enough higher priority requests are active. In the rotating mode, any request will receive service within a maximum of seven other service cycles no matter what pattern the request inputs follow.

Mode bit 1 selects the individual/common vector option. Individual vectoring provides a unique location in the response memory for each interrupt request. The common vector option always supplies the response associated with $\overline{IREQ0}$ no matter which request is being acknowledged.

Mode bit 2 specifies interrupt or polled operation. In the polled mode the Group Interrupt output is disabled. The CPU may read the Status register to determine if a request is pending. Since \overline{IACK} pulses are not normally supplied in polled mode, the \overline{IRR} bit is not automatically cleared, but may be cleared by command. With no \overline{IACK} input the ISR and the response memory are not used. An Am9519A in the polled mode has \overline{EI} connected to \overline{EO} so that in multichip interrupt systems the polled chip is functionally removed from the priority hierarchy.

Mode bit 3 specifies the sense of the \overline{GINT} output. When active high polarity is selected the output is a two-state configuration. For active low polarity, the output is open drain and requires an external pull-up resistor to provide the high logic level. The open drain output allows wired-or configurations with other similar output signals.

Mode bit 4 specifies the sense of the \overline{IREQ} inputs. When active low polarity is selected, the \overline{IRR} responds to falling edges on the request inputs. When active high is selected, the \overline{IRR} responds to rising edges.

Mode bits 5 and 6 specify the register that will be read on subsequent data read operations ($\overline{C/D} = 0$, $\overline{RD} = 0$). This preselection remains valid until changed by a reset or a command.

Mode bit 7 is the master mask bit that disables all request inputs. It is used to disable all interrupts without modifying the IMR so that the previous IMR contents are valid when interrupts are re-enabled. When the master mask bit is low, it causes the \overline{EO} line to remain disabled (low). Thus, for multiple-chip interrupt systems, one master mask bit can disable the whole interrupt structure. Alternatively, portions of the structure may be disabled. The state of the master mask bit is available as bit S3 of the Status register.

Programming

After reset, the Am9519A must be initialized by the CPU in order to perform useful work. At a minimum, the master mask bit and at least one of the IMR bits should be enabled. If vectoring is to be used, the response memory must be loaded; if not, the mode must be changed to a non-vector mode. Normally, the first step will be to modify the Mode register and the Auto clear register in order to establish the configuration desired for the application. Then the response memory and byte count will be loaded for those request levels that will be in use. Finally, the master mask bit and at least portions of the IMR will be enabled to allow interrupt processing to proceed.

Commands

The host CPU configures, changes and inspects the internal condition of the Am9519A using the set of commands shown in Figure 4. An "X" entry in the table indicates a "don't care" state. All commands are entered by directly loading the Command register as shown in Figure 3 ($C/\bar{D} = 1$, $WR = 0$). Figure 5 shows the coding assignments for the Byte Count registers. A detailed description of each command is contained in the Am9519A Application Note AMPUB-071.

BY1	BY0	COUNT
0	0	1
0	1	2
1	0	3
1	1	4

Figure 5. Byte Count Coding.

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Reset
0	0	0	1	0	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0–4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Preselect IMR for subsequent loading from data bus
1	1	0	0	X	X	X	X	Preselect Auto Clear register for subsequent loading from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count register and preselect response memory level specified by L2, L1, L0 for subsequent loading from data bus

Figure 4. Am9519A Command Summary.

Am9519A

MAXIMUM RATINGS above which useful life may be impaired

Storage Temperature	−65°C to +150°C
Ambient Temperature Under Bias	−55°C to +125°C
VCC with Respect to VSS	−0.5V to +7.0V
All Signal Voltages with Respect to VSS	−0.5V to +7.0V
Power Dissipation (Package Limitation)	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	Ambient Temperature	VCC	VSS
Am9519ADC/CC Am9519A-1DC	0°C ≤ T _A ≤ +70°C	+5.0V ±5%	0V
Am9519ADM	−55°C ≤ T _A ≤ +125°C	+5.0V ±10%	0V

ELECTRICAL CHARACTERISTICS Over Operating Range (Note 1)

Parameter	Description	Test Conditions	Min.	Typ.	Max.	Unit
VOH	Output High Voltage (Note 12)	IOH = −200μA	2.4			Volts
		IOH = −100μA (EO only)	2.4			
VOL	Output Low Voltage	IOL = 3.2mA			0.4	Volts
		IOL = 1.0mA (EO only)			0.4	
VIH	Input High Voltage		2.0		VCC	Volts
VIL	Input Low Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VIN ≤ VCC	EI Input	−60	10	μA
			Other Inputs	−10	10	
IOZ	Output Leakage Current	VSS ≤ VOUT ≤ VCC, Output off	−10		10	μA
ICC	VCC Supply Current	T _A = +25°C		80	125	mA
		T _A = 0°C		100	145	
CO	Output Capacitance	fc = 1.0MHz			15	pF
CI	Input Capacitance	T _A = 25°C			10	
CIO	I/O Capacitance	All pins at 0V			20	

SWITCHING CHARACTERISTICS Over Operating Range (Notes 2, 3, 4, 5)

Parameters	Description	Am9519A		Am9519A-1		Units
		Min	Max	Min	Max	
TAVRL	C/ \bar{D} Valid and \bar{CS} LOW to Read LOW	0		0		ns
TAVWL	C/ \bar{D} Valid and \bar{CS} LOW to Write LOW	0		0		ns
TCLPH	\bar{RIP} LOW to \bar{PAUSE} HIGH (Note 6)	75	300	75	300	ns
TCLQV	\bar{RIP} LOW to Data Out Valid (Note 7)		50		40	ns
TDVWH	Data In Valid to Write HIGH	250		200		ns
TEHCL	Enable in HIGH to \bar{RIP} LOW (Notes 8, 9)	30	300	30	300	ns
TIVGV	Interrupt Request Valid to Group Interrupt Valid		800		650	ns
TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)	250		250		ns
TKHCH	\bar{IACK} HIGH to \bar{RIP} HIGH (Note 8)		400		350	ns
TKHKL	\bar{IACK} HIGH to \bar{IACK} LOW (\bar{IACK} Recovery)		500		500	ns
TKHNH	\bar{IACK} HIGH to EO HIGH (Notes 10, 11)		800		700	ns
TKHQX	\bar{IACK} HIGH to Data Out Invalid	20	200	20	100	ns
TKLCL	\bar{IACK} LOW to \bar{RIP} LOW (Note 8)	75	600	75	450	ns
TKLKH	\bar{IACK} LOW to \bar{IACK} HIGH (1st \bar{IACK})	975		800		ns
TKLNL	\bar{IACK} LOW to EO LOW (Notes 10, 11)		125		100	ns
TKLPL	\bar{IACK} LOW to \bar{PAUSE} LOW	25	175	25	125	ns
TKLQV	\bar{IACK} LOW to Data Out Valid (Note 7)	25	300	25	200	ns
TKLQV1	1st \bar{IACK} LOW to Data Out Valid	75	650	75	490	ns
TPHKH	\bar{PAUSE} HIGH to \bar{IACK} HIGH	0		0		ns
TRHAX	Read HIGH to C/ \bar{D} and \bar{CS} Don't Care	0		0		ns
TRHQX	Read HIGH to Data Out Invalid	20	200	20	100	ns
TRLQV	Read LOW to Data Out Valid		300		200	ns
TRLQX	Read LOW to Data Out Unknown	50		50		ns
TRLRH	Read LOW to Read HIGH (\bar{RD} Pulse Duration)	300		250		ns
TWHAX	Write HIGH to C/ \bar{D} and \bar{CS} Don't Care	0		0		ns
TWHDX	Write HIGH to Data In Don't Care	0		0		ns
TWHRW	Write HIGH to Read or Write LOW (Write Recovery)	600		400		ns
TWLWH	Write LOW to Write HIGH (\bar{WR} Pulse Duration)	300		250		ns

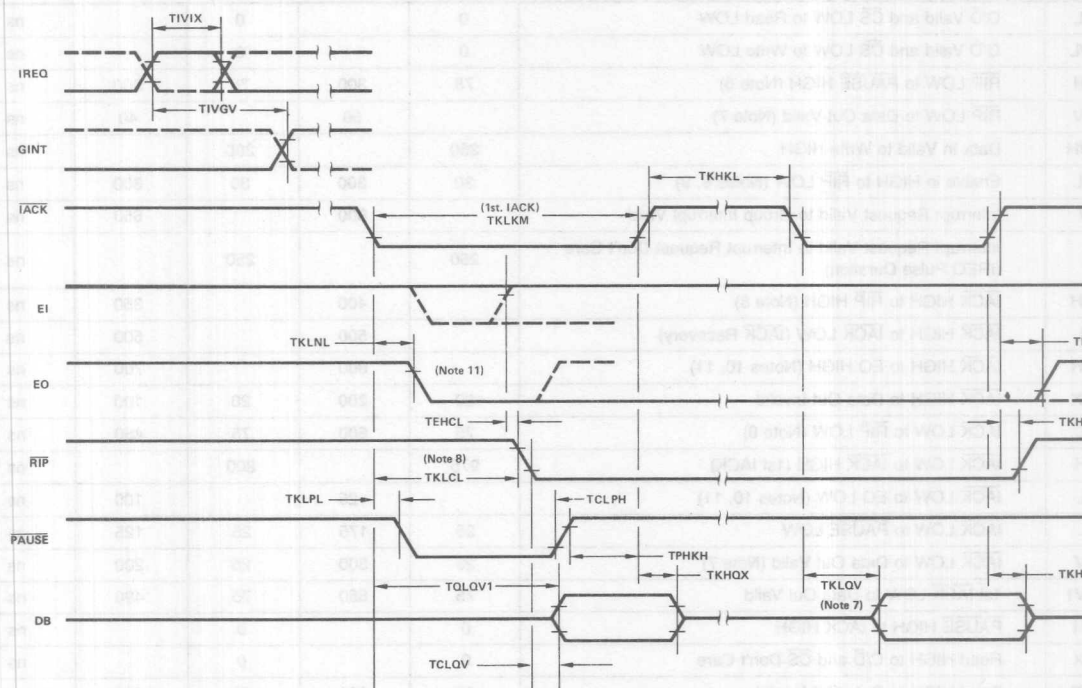
NOTES:

- Typical values for $T_A = 25^\circ\text{C}$, nominal supply voltage and nominal processing parameters.
- Test conditions assume transition times of 20ns or less, timing reference levels of 0.8V and 2.0V and output loading of one TTL gate plus 100pF, unless otherwise noted.
- Transition abbreviations used for the switching parameter symbols include: H = High, L = Low, V = Valid, X = unknown or don't care, Z = high impedance.
- Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address (\bar{CS} and C/ \bar{D}), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = \bar{RIP} .
- Switching parameters are listed in alphabetical order.
- During the first \bar{IACK} pulse, \bar{PAUSE} will be low long enough to allow for priority resolution and will not go high until after \bar{RIP} goes low (TCLPH).
- TKLQV applies only to second, third and fourth \bar{IACK} pulses while \bar{RIP} is low. During the first \bar{IACK} pulse, Data Out will be valid following the falling edge of \bar{RIP} (TCLQV).
- \bar{RIP} is pulled low to indicate that an interrupt request has been selected. \bar{RIP} cannot be pulled low until EI is high following an internal delay. TKLCL will govern the falling edge of \bar{RIP} when

EI is always high or is high early in the acknowledge cycle. TEHCL will govern when EI goes high later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain. \bar{RIP} remains low until after the rising edge of the \bar{IACK} pulse that transfers the last response byte for the selected IREQ.

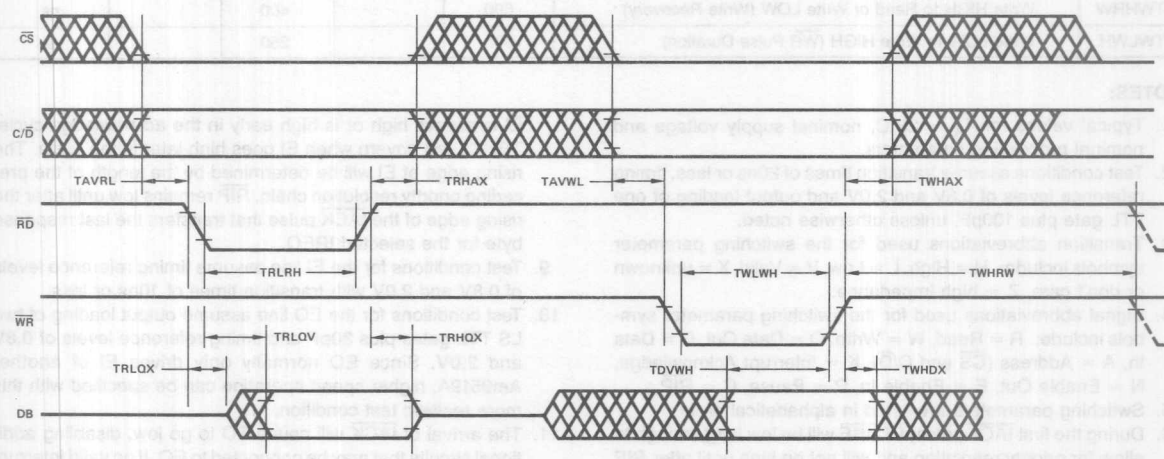
- Test conditions for the EI line assume timing reference levels of 0.8V and 2.0V with transition times of 10ns or less.
- Test conditions for the EO line assume output loading of two LS TTL gates plus 30pF and timing reference levels of 0.8V and 2.0V. Since EO normally only drives EI of another Am9519A, higher speed operation can be specified with this more realistic test condition.
- The arrival of \bar{IACK} will cause EO to go low, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return high when EI is high. If a pending request is selected, EO will stay low until after the last \bar{IACK} pulse for that interrupt is complete and \bar{RIP} goes high.
- VOH specifications do not apply to \bar{RIP} or to \bar{GINT} when active-low. These outputs are open-drain and VOH levels will be determined by external circuitry.

SWITCHING WAVEFORMS



MOS-144

Interrupt Operations



MOS-145

Data Bus Transfers

APPLICATIONS

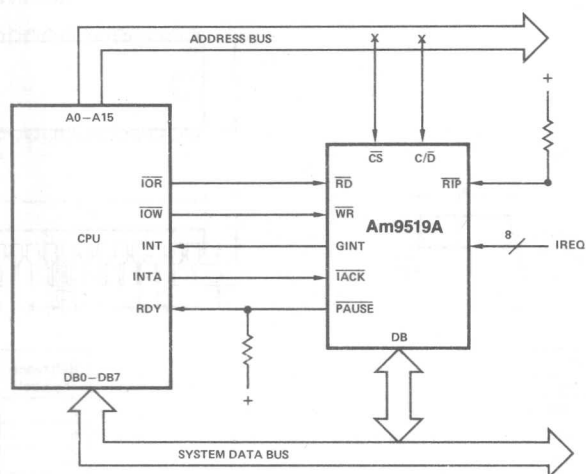


Figure 6. Base Interrupt System Configuration.

MOS-146

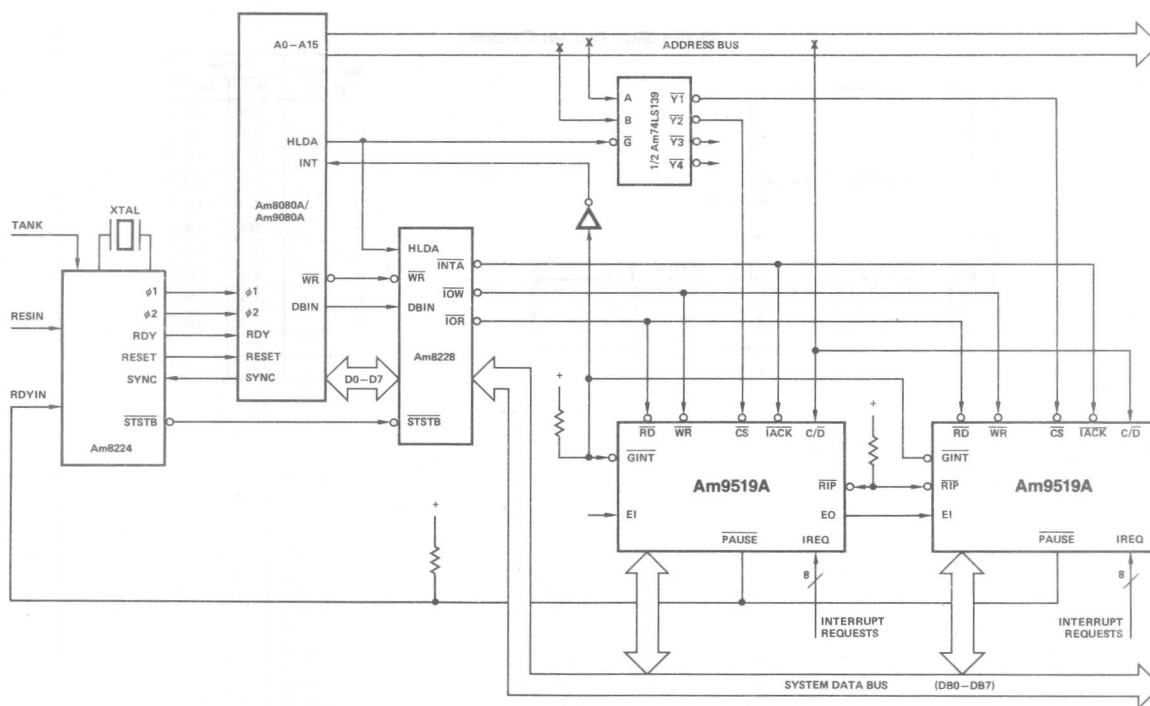
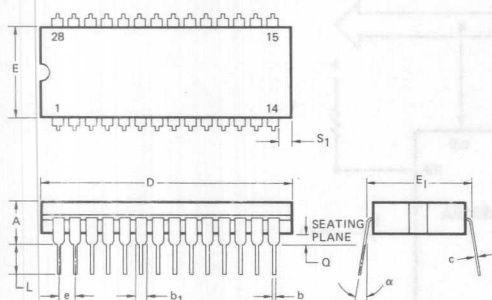


Figure 7. Expanded Interrupt System Configuration.

MOS-147

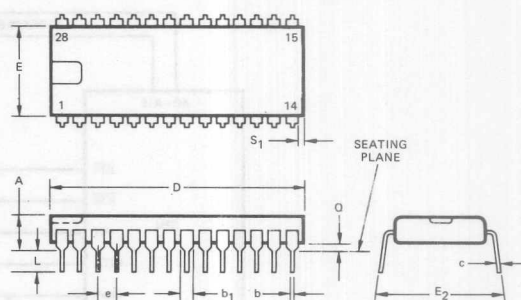
PHYSICAL DIMENSIONS

28-Pin Cerdip



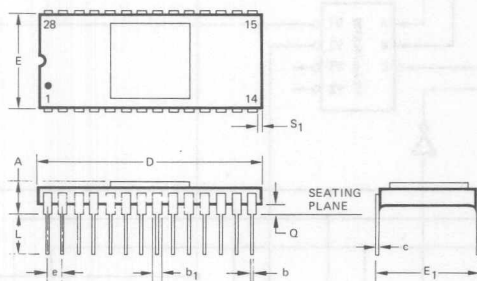
Reference Symbol	Inches	
	Min.	Max.
A	.150	.225
b	.016	.020
b ₁	.045	.065
c	.009	.012
D	1.440	1.490
E	.510	.545
E ₁	.600	.620
e	.090	.110
L	.125	.150
Q	.015	.060
S ₁	.010	

28-Pin Molded DIP



Reference Symbol	Inches	
	Min.	Max.
A	.150	.200
b	.015	.020
b ₁	.055	.065
c	.009	.011
D	1.450	1.440
E	.530	.550
E ₂	.585	.700
e	.090	.110
L	.125	.160
Q	.015	.060
S ₁	.040	.070

28-Pin Side-Brazed Ceramic



Reference Symbol	Inches	
	Min.	Max.
A	.100	.200
b	.015	.022
b ₁	.030	.060
c	.008	.013
D	1.380	1.420
E	.560	.600
E ₁	.580	.620
e	.090	.110
L	.120	.160
Q	.020	.060
S ₁ *	.005	
alpha	0	0

Microprocessors

AmZ8002

16-Bit Microprocessor

PRELIMINARY DATA

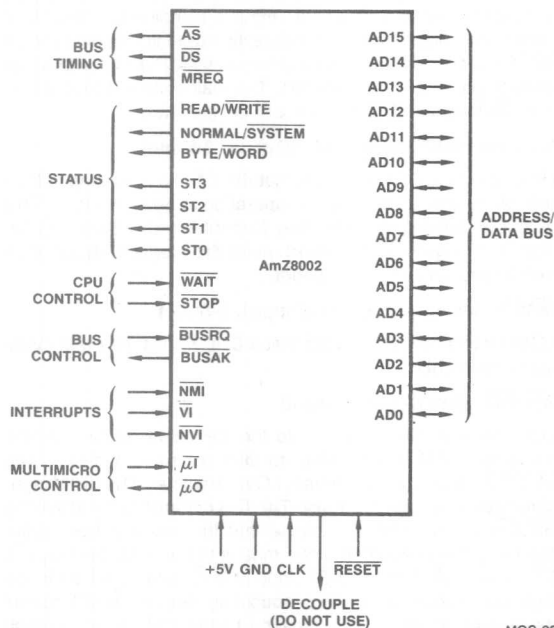
DISTINCTIVE CHARACTERISTICS

- Sixteen general purpose registers
- Direct addressing up to 64KB memory
- Software compatible with AmZ8001 microprocessor
- Powerful instructions with flexible addressing modes
- Privileged/Non-Privileged mode of operation
- Sophisticated interrupt structure
- On-chip memory refresh facility
- TTL compatible inputs and outputs
- Single phase clock
- Single +5V power supply
- 40-pin package

GENERAL DESCRIPTION

The AmZ8002 is a general-purpose 16-bit CPU belonging to the AmZ8000 family of microprocessors. Its architecture is centered around sixteen 16-bit general registers. The CPU deals with 16-bit address spaces and hence can address directly 64 Kilo-bytes of memory. Facilities are provided to maintain three distinct address spaces – code, data and stack. The AmZ8002 implements a powerful instruction set with flexible addressing modes. These instructions operate on several data types – bit, byte, word (16-bit), long word (32-bit), byte string and word string. The CPU can execute instructions in one of two modes – System and Normal. Sometimes these modes are also known as Privileged and Non-Privileged, respectively. The CPU also contains an on-chip memory refresh facility. The AmZ8002 is software compatible with the AmZ8001 microprocessor. The AmZ8002 is fabricated using silicon-gate N-MOS technology and is packaged in a 40-pin DIP. The AmZ8002 requires a single +5 power supply and a single phase clock for its operation.

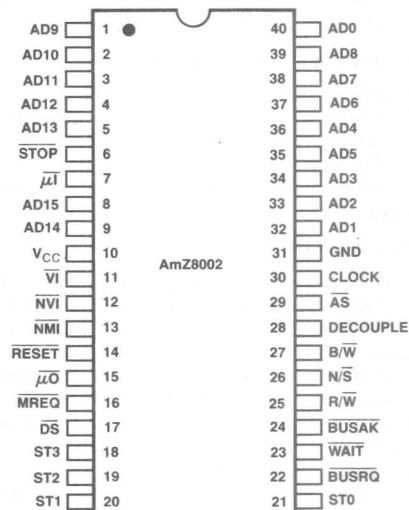
LOGIC SYMBOL



MOS-225

CONNECTION DIAGRAM

Top View



Note: Pin 1 is marked for orientation.

MOS-226

ORDERING INFORMATION

Package Type	Ambient Temperature	Maximum Clock Frequency
		4MHz
Hermetic DIP	$0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$	AmZ8002DC

AD0-AD15: Address/Data Bus (Bidirectional, 3-State)

This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 the most significant. The \overline{AS} output and \overline{DS} output will indicate whether the bus is used for address or data. The status output lines ST0-ST3 will indicate the type of transaction; memory or I/O.

\overline{AS} : Address Strobe (Output, 3-State)

LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the \overline{AS} output (see timing diagrams). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.

\overline{DS} : Data Strobe (Output, 3-State)

LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/W output indicates the direction of data transfer — read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus when \overline{DS} goes LOW. A LOW-to-HIGH transition on the \overline{DS} output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the \overline{DS} output indicates that data is setup on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the \overline{DS} output (see timing diagram).

R/W: Read/Write (Output, 3-State)

This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as \overline{AS} going LOW and remains stable for the duration of the whole transaction (see timing diagram).

B/W: Byte/Word (Output, 3-State)

This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as \overline{AS} going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port and B/W determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.

ST0-ST3: Status (Outputs, 3-State)

These four outputs contain information regarding the current transaction in a coded form. The status line codes are shown in the following table:

L	L	L	H	Memory Refresh
L	L	H	L	Normal I/O Transaction
L	L	H	H	Special I/O Transaction
L	H	L	L	Reserved
L	H	L	H	Non-Maskable Interrupt Acknowledge
L	H	H	L	Non-Vectored Interrupt Acknowledge
L	H	H	H	Vectored Interrupt Acknowledge
H	L	L	L	Memory Transaction for Operand
H	L	L	H	Memory Transaction for Stack
H	L	H	L	Reserved
H	L	H	H	Reserved
H	H	L	L	Memory Transaction for Instruction Fetch (Subsequent Word)
H	H	L	H	Memory Transaction for Instruction Fetch (First Word)
H	H	H	L	Reserved
H	H	H	H	Reserved

WAIT: Wait (Input)

LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer and hence the current transaction should be stretched. The WAIT input is sampled by the CPU at certain instances during the transaction (see timing diagram). If WAIT input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the WAIT input is HIGH at the sampling instant.

N/S: Normal/System Mode (Output, 3-State)

HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the program status information section of this document.

MREQ: Memory Request (Output, 3-State)

LOW on this output indicates that a CPU transaction with memory is taking place.

BUSRQ: Bus Request (Input)

LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The BUSRQ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating BUSAK output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, \overline{AS} , \overline{DS} , B/W, R/W, N/S, ST0-ST3 and MREQ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The BUSRQ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the BUSAK output LOW. After completing the transactions, the device must disable the AD0-AD15, \overline{AS} , \overline{DS} , B/W, R/W, N/S, ST0-ST3 and MREQ into the high impedance state and stop driving the BUSRQ input LOW. The CPU will make BUSAK output HIGH sometime later and take back the bus control.

BUSAK: Bus Acknowledge (Output)

LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.

NMI: Non-Maskable Interrupt (Input)

HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the NMI can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.

VI: Vectored Interrupt (Input)

LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The VIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The VI input can be driven LOW any time and should be held LOW until acknowledged.

NVI: Non-Vectored Interrupt (Input)

LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The NVI input can be driven LOW anytime and should be held LOW until acknowledged.

 μ I: Micro-In (Input)

This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.

 μ O: Micro-Out (Output)

This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.

RESET: Reset (Input)

LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.

CLK: Clock (Input)

All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.

DECOUPLE: (Output)

Output from the on-chip substrate bias generator. Do not use.

STOP: Stop (Input)

This active LOW input facilitates one instruction at a time operation. See the section on single stepping.

PROCESSOR ORGANIZATION

The following is a brief discussion of the AmZ8002 CPU. For detailed information, see the AmZ8001/AmZ8002 Processor Instruction Set Manual (Publication No. AM-PUB086).

General Purpose Registers

The CPU is organized around sixteen 16-bit general purpose registers R0 through R15 as shown in Figure 1. For byte operations, the first eight registers (R0 through R7) can also be addressed as sixteen 8-bit registers designated as RL0, RH0 and so on to RL7 and RH7. The sixteen registers can also be grouped in pairs RR0, RR2 and so on to RR14 to form eight long

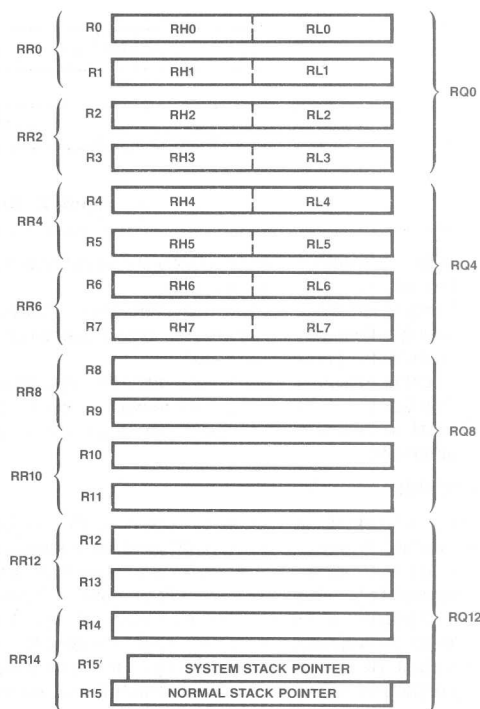


Figure 1. AmZ8002 General Registers.

MOS-227

word (32-bit) registers. Similarly, the sixteen registers can be grouped in quadruples RQ0, RQ4, RQ8 and RQ12 to form four 64-bit registers.

STACK POINTER

The AmZ8002 architecture allows stacks to be maintained in the memory. Any general purpose register except R0 can be used as a stack pointer in stack manipulating instructions such as PUSH and POP. However, certain instructions such as sub-routine call and return make implicit use of the register R15 as the stack pointer. Two implicit stacks are maintained – normal stack using R15 as the stack pointer and system stack using R15' as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, R15 is active, and if the CPU is in System Mode R15' will be used instead of R15. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.

PROCESSOR STATUS

The CPU status consists of the 16-bit Program Counter (PC) and the 16-bit Flag and Control Word (FCW) register (see Figure 2). The following is a brief description of the FCW bits.

- S/N:** System/Normal – 1 indicates System Mode and 0 indicates Normal Mode.
- VIE:** Vectored Interrupt Enable – 1 indicates that Vectored Interrupt requests will be honored.
- NVIE:** Non-Vectored Interrupt Enable – 1 indicates that non-vectored interrupt requests will be honored.
- C:** Carry – 1 indicates that a carry has occurred from the most significant bit position when performing arithmetic operations.
- Z:** Zero – 1 indicates that the result of an operation is zero.

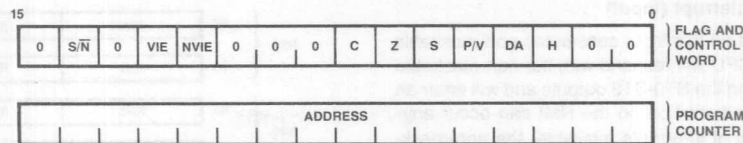


Figure 2. AmZ8002 Processor Status.

MOS-228

- S:** Sign – 1 indicates that the result of an operation is negative i.e., most significant bit is one.
- P/V:** Parity/Overflow – 1 indicates that there was an overflow during arithmetic operations. For logical operations this bit indicates parity of the result.
- DA:** Decimal Adjust – Records byte arithmetic operations.
- H:** Half Carry – 1 indicates that there was a carry from the most significant bit of the lower digit during byte arithmetic.

DATA TYPES

The AmZ8002 instructions operate on bits, digits (4 bits), bytes (8 bits), words (16 bits), long words (32 bits), byte strings and word strings type operands. Bits can be set, reset or tested. Digits are used to facilitate BCD arithmetic operations. Bytes are used for characters and small integers. Words are used for integer values and addresses while long words are used for large integer values. All operands except strings can reside either in memory or general registers. Strings can reside in memory only.

INTERRUPT AND TRAP STRUCTURE

Interrupt is defined as an external asynchronous event requiring program interruption. For example, interruption is caused by a peripheral needing service. Traps are synchronous events resulting from execution of certain instructions under some defined circumstances. Both interrupts and traps are handled in a similar manner by the AmZ8002.

The AmZ8002 supports three types of interrupts in order of descending priority – non-maskable, vectored and non-vectored. The vectored and non-vectored interrupts can be disabled by appropriate control bits in the FCW. The AmZ8002 has three traps – system call, unimplemented opcode and privileged instruction. The traps have higher priority than interrupts.

When an interrupt or trap occurs, the current program status is automatically pushed on to the system stack. The program status consists of processor status (i.e., PC and FCW) plus a 16-bit identifier. The identifier contains the reason, source and other coded information relating to the interrupt or trap.

After saving the current program status, the new processor status is automatically loaded from the new program status area located in the memory. This area is designated by the New Program Status Area Pointer (NPSAP) register. See AM-PUB086 publication for further details.

ADDRESSING MODES

Information contained in the AmZ8002 instructions consists of the operation to be performed, the operand type and the location of the operands. Operand locations are designated by general register addresses, memory addresses or I/O addresses. The addressing mode of a given instruction defines the address space referenced and the method to compute the operand address. Addressing modes are explicitly specified or implied in an instruction. Figure 3 illustrates the eight explicit addressing modes: Register (R), Immediate (IM), Indirect Register (IR), Direct Address (DA), Indexed (X), Relative Address (RA), Base Address (BA) and Base Indexed (BX).

INPUT/OUTPUT

A set of I/O instructions are provided to accomplish byte or word transfers between the AmZ8002 and I/O devices. I/O devices are addressed using 16-bit I/O port addresses and I/O address space is not a part of the memory address space. Two types of I/O instructions are provided; each with its own 16-bit address space. I/O instructions include a comprehensive set of In, Out and Block transfers.

CPU TIMING

The AmZ8002 accomplishes instruction execution by stepping through a pre-determined sequence of machine cycles, such as memory read, memory write, etc. Each machine cycle requires between three and ten clock cycles. Bus Requests by DMA devices are granted at machine cycle boundaries. No machine cycle is longer than ten clock cycles; thus assuring fast response to a Bus Request (assuming no extra wait states). The start of a machine cycle is always marked by a LOW pulse on the \overline{AS} output. The status output lines ST0-ST3 indicate the nature of the current cycle in a coded form.

STATUS LINE CODES

Status line coding was listed in the table shown under ST0-ST3 outputs in the Interface Signal Description. The following is a detailed description of the status codes.

Internal Operation:

This status code indicates that the AmZ8002 is going through a machine cycle for its internal operation. Figure 4 depicts an internal operation cycle. It consists of three clock periods identified as T1, T2 and T3. The \overline{AS} output will be activated with a LOW pulse by the AmZ8002 to mark the start of a machine cycle. The ST0-ST3 will reflect the code for the internal operation. The \overline{MREQ} , \overline{DS} and R/\overline{W} outputs will be HIGH. The N/\overline{S} output will remain at the same level as in the previous machine cycle. The AmZ8002 will ignore the \overline{WAIT} input during the internal operation cycle. The CPU will drive the AD0-AD15 bus with unspecified information during T1. However, the bus will go into high impedance during T2 and remain in that state for the remainder of the cycle. The B/\overline{W} output is also activated by the CPU with unspecified information.

Memory Refresh:

This status code indicates that AmZ8002 is accessing the memory to refresh. The refresh cycle consists of three clock periods as depicted in Figure 5. The CPU will activate the \overline{AS} output with a LOW pulse to mark the beginning of a machine cycle and ST0-ST3 outputs will reflect the refresh cycle code. The least significant 9 lines of the AD0-AD15 bus contain the refresh address. Because the memory is word organized, the AD0 will always be LOW. The most significant 7 bus lines are not specified. The \overline{DS} output will remain HIGH for the entire cycle while R/\overline{W} , B/\overline{W} and N/\overline{S} outputs will remain at the same level as in the machine cycle prior to refresh. The AD0-AD15 bus will go into high impedance state during T2 period and remain there for





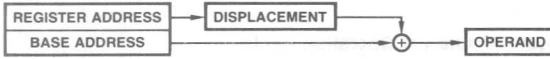
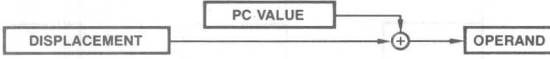
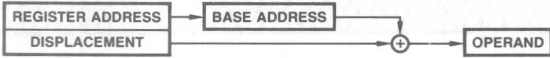

Mode	Operand Addressing		Operand Value
	In the Instruction	In a Register	
Register			The content of the register.
Immediate			In the instruction
Indirect Register			The content of the location whose address is in the register.
Direct Address			The content of the location whose address is in the instruction.
Index			The content of the location whose address is the address in the instruction, offset by the content of the working register.
Relative Address			The content of the location whose address is the content of the program counter, offset by the displacement in the instruction.
Base Address			The content of the location whose address is the address in the register, offset by the displacement in the instruction.
Base Index			The content of the location whose address is the address in the register, offset by the displacement in the register.

Figure 3. Addressing Modes.

MOS-229

the remainder of the cycle. The AmZ8002 will activate the MREQ output LOW during the refresh cycle. It should be noted that WAIT input is ignored by the CPU for refresh operations.

I/O Transactions:

There are two status line codes used for I/O transaction cycles. The AmZ8002 provides two separate I/O spaces and two types of instructions called Normal I/O and Special I/O. Each I/O space is addressed by a 16-bit address called port address. The timing for both types of I/O transactions is essentially identical. A typical I/O cycle consists of four clock periods T1, T2, TWA and T3 as shown in Figure 6. The TWA is the wait state; insertion of one wait state for an I/O cycle is always automatic. Additional

wait cycles can be inserted by LOW on the WAIT input. The WAIT input is sampled during every TW state. If this input is LOW, one more wait state will be inserted. Insertion of wait states continues until WAIT input is HIGH. T3 state will follow the last wait state to complete the I/O cycle.

During I/O cycles the ST0-ST3 outputs will reflect appropriate code depending on the type of instruction being executed (Normal I/O or Special I/O). AS output will be pulsed LOW to mark the beginning of the cycle. The CPU drives the AD0-AD15 bus with the 16-bit port address specified by the current instruction. The N/S output will be LOW indicating that CPU is operating in the system mode. It should be recalled that the N/S output is derived from the appropriate bit in the FCW register. All I/O instructions are privileged instructions and will be allowed to

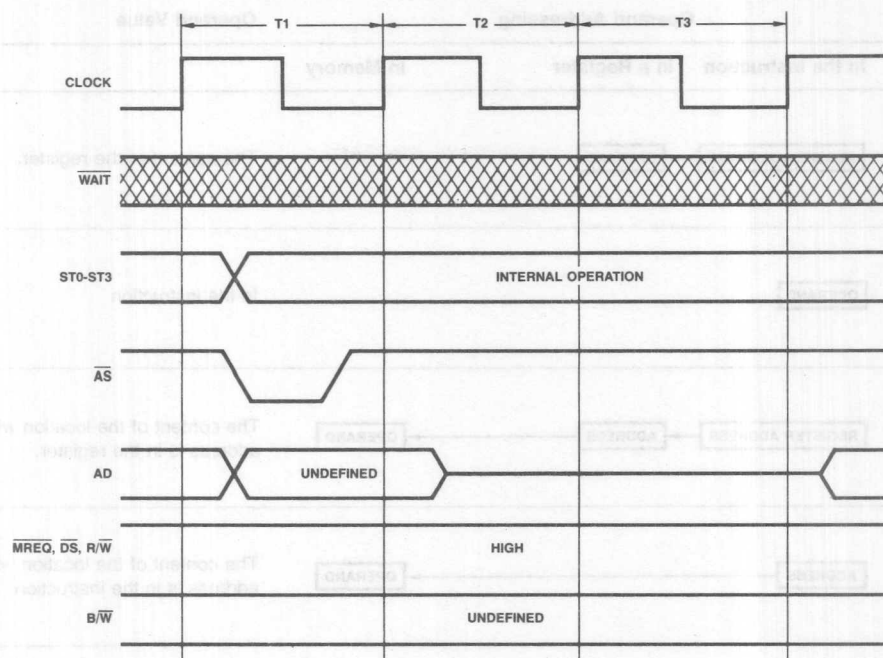


Figure 4. Internal Operation Cycle.

MOS-230

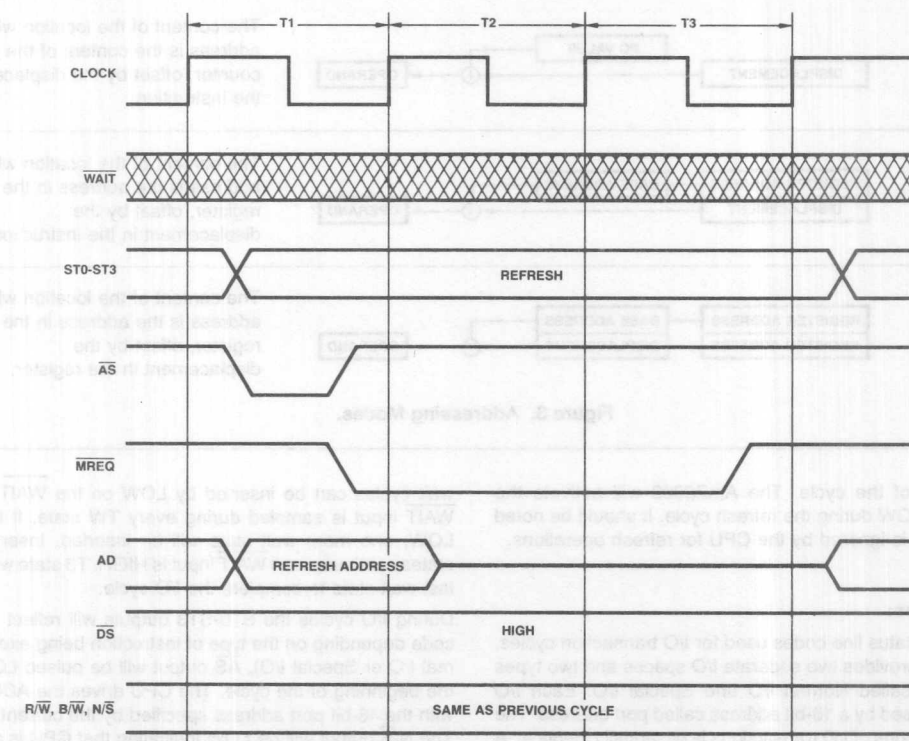


Figure 5. Refresh Cycle.

MOS-231

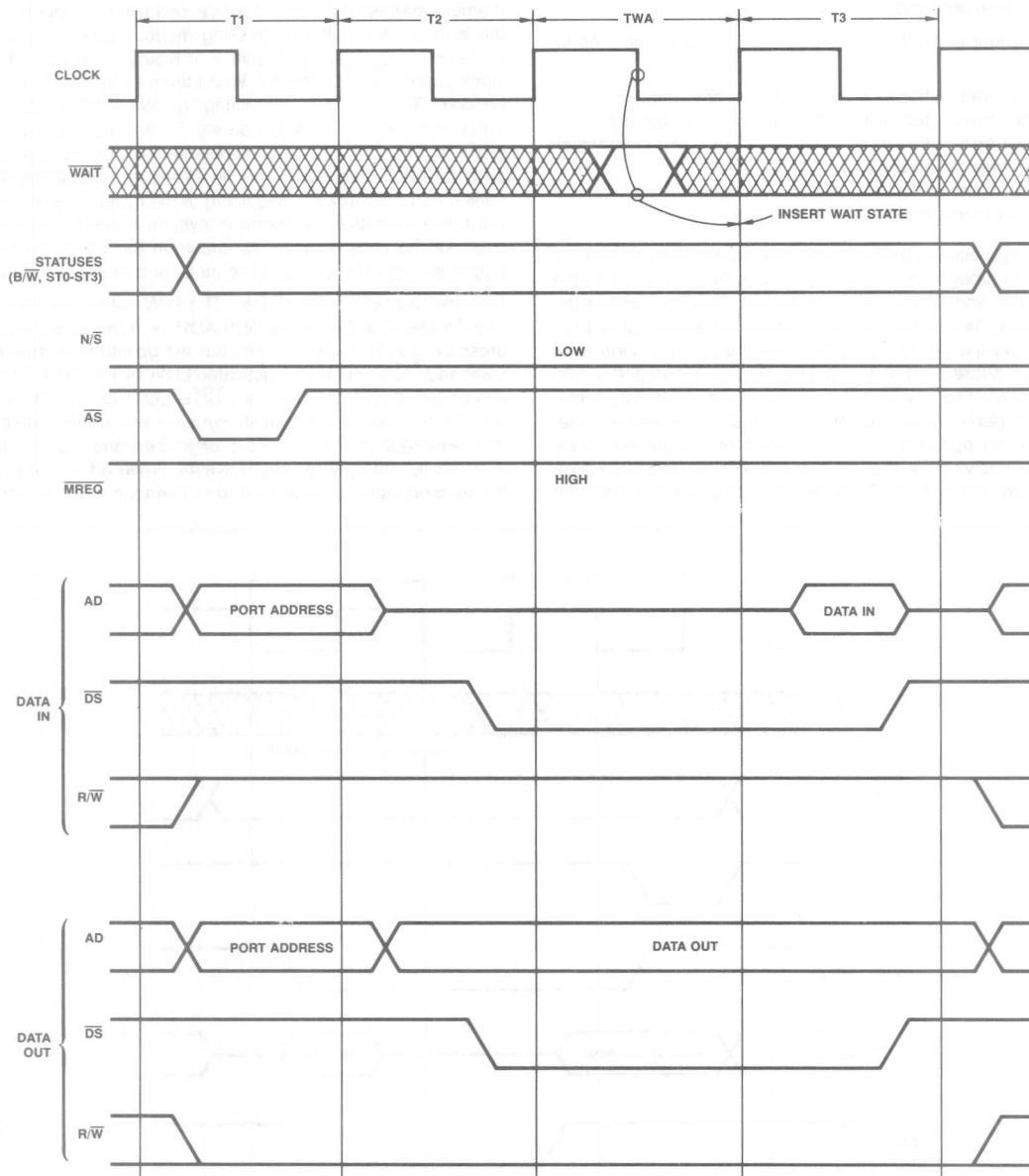


Figure 6. AmZ8002 I/O Cycle.

MOS-232

execute only if the FCW specifies system mode operation. The MREQ output will be HIGH. The AmZ8002 I/O instructions provide both word or byte transactions. The B/W output will be HIGH or LOW depending whether the instruction specifies a byte or word transfer.

Two kinds of I/O transfers should be considered: Data In means reading from the device and Data Out means writing into the device. For In operations, the R/W output will be HIGH. The AD0-AD15 bus will go into high impedance state during T2. During byte input instructions, the CPU reads either the even or odd half of the Data Bus, dependent upon the port address. If the port address is even, the most significant half of the Data Bus is read. If the port address is odd, the least significant half of

the Data Bus is read. During word input instructions, the CPU reads all 16 bits of the Data Bus. The AmZ8002 will drive the DS output LOW to signal to the device that data can be gated on to the bus. The CPU will accept the data during T3 and DS output will go HIGH signalling the end of an I/O transaction.

For Data Out, the R/W output will be LOW. The AmZ8002 will provide data on the AD0-AD15 bus and activates the DS output LOW during T2. During byte output instructions, the CPU duplicates the byte data onto both the high and low halves of the Data Bus and external logic, using A0, enables the appropriate byte port. During word output instructions the CPU outputs data onto all 16 bits of the Data Bus. The DS output goes HIGH during T3 and the cycle is complete.

Memory Transactions:

There are four status line codes that indicate a memory transaction:

- Memory transaction to read or write an operand
- Memory transaction to read from or write into the stack
- Memory transaction to fetch the first word of an instruction (sometimes called IF1)
- Memory transaction to fetch the subsequent word of an instruction (sometimes called IFN).

It can be appreciated that all the above transactions essentially fall into two categories: memory read and memory write. In the case of IF1 and IFN cycles, the memory will be read at the address supplied by the program counter. All AmZ8002 instructions are multiples of 16-bit words. Words are always addressed by an even address. Thus IF1 and IFN cycles involve performing a memory read for words. On the other hand, a memory transaction for operand and stack operation could be a read or write. Moreover, an operand could be a word or a byte. For stack operation involving the implied stack pointer the address will be supplied by the R15 (or R15'). For operand transactions, the

memory address will come from several sources depending on the instruction and the addressing mode. Memory transaction cycle timing is shown in Figure 7. It typically consists of three clock periods T1, T2 and T3. Wait states (TW) can be inserted between T2 and T3 by activating the WAIT input LOW. The WAIT input will be sampled during T2 and during every subsequent TW. The ST0-ST3 outputs will reflect the appropriate code for the current cycle early in T1 and the \overline{AS} output will be pulsed LOW to mark the beginning of the cycle. The N/S output will indicate whether the normal or system address space will be used for the current cycle. As shown in the figure the MREQ output will go LOW during T1 to indicate a memory operation.

Consider a read operation first. The R/W output will be HIGH. The AmZ8002 will drive the AD0-AD15 with the appropriate address early in T1. During T2, the bus will go into high impedance state and \overline{DS} output will be activated LOW by the CPU. The data can be gated on to the bus when \overline{DS} is LOW. During T1 the B/W will also be activated to indicate byte or word will be transacted. The AmZ8002 memory is word organized and words are addressed by even addresses. However, when addressing bytes, the memory address may be odd or even; an even address for

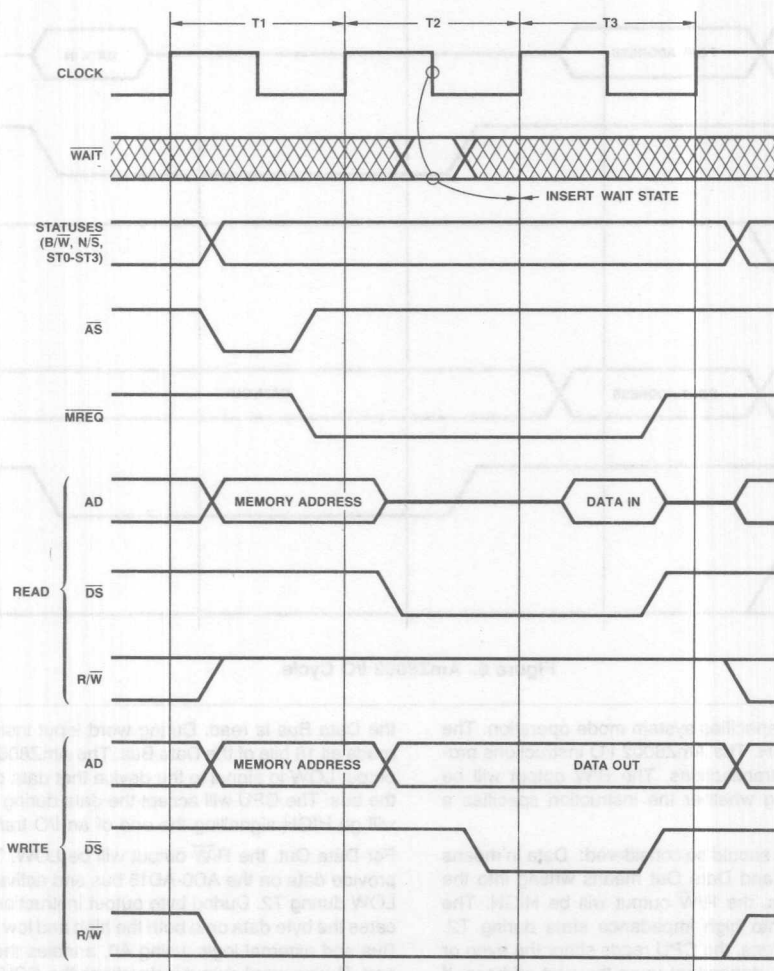


Figure 7. Memory Transactions.

MOS-233

most significant byte of a word and the next odd address for the least significant byte of that word. When reading a byte from the memory, the least significant address bit can be ignored and the whole word containing the desired byte is gated on to the bus. The CPU will pick the appropriate byte automatically. The AmZ8002 will drive the \overline{DS} output HIGH indicating data acceptance.

Consider the write operation next. The R/\overline{W} output will be LOW. The AmZ8002 removes the address and gates out the data to be written on the bus and activates the \overline{DS} output LOW during T2. If the data to be written is a byte then the same byte will be on both halves of the bus. The \overline{DS} output will go HIGH during T3 signifying completion of the cycle.

Interrupt Acknowledge:

There are three status line codes devoted to interrupt acknowledgement. These correspond to non-maskable, vectored and non-vectored interrupts. The Interrupt Acknowledge cycle is illustrated in Figure 8. The NMI input of the AmZ8002 is edge detected i.e., a HIGH to LOW input level change is stored in an internal latch. Similar internal storage is not provided for the \overline{VI} and \overline{NVI} inputs. For \overline{VI} and \overline{NVI} inputs to cause an interruption, the corresponding interrupt enable bits in the FCW must be 1. For the following discussion, both the VIE and NVIE bits in the FCW are assumed to be 1.

As shown in the figure, the \overline{VI} input, \overline{NVI} input and the internal NMI latch output are sampled during T3 of the last machine cycle of an instruction.

A LOW on these signals triggers the corresponding interrupt acknowledge sequence described below. The AmZ8002 executes a dummy IF1 cycle prior to entering the actual acknowledge cycle (see memory transactions for IF1 cycle description). During this dummy IF1 cycle, the program counter is not updated; instead the implied system stack pointer ($R15'$) will be decremented. Following the dummy IF1 cycle is the actual interrupt acknowledge cycle.

The interrupt acknowledge cycle typically consists of 10 clock periods; T1 through T5 and five automatic TW (wait) states. As usual, the \overline{AS} output will be pulsed LOW during T1 to mark the beginning of a cycle. The ST0-ST3 outputs will reflect the appropriate interrupt acknowledge code, the MREQ output will be HIGH, the N/\overline{S} output remains the same as in the preceding cycle, the R/\overline{W} output will be HIGH and the B/\overline{W} output will be LOW. The AmZ8002 will drive the AD0-AD15 bus with unspecified information during T1 and the bus will go into the high impedance state during T2. Three TWA states will automatically follow T2. The WAIT input will be sampled during the third TWA state.

If LOW, an extra TW state will be inserted and the WAIT will be sampled again during TW. Such insertion of TW states continues until the WAIT input is HIGH. After the last TW state, the \overline{DS} output will go LOW and two more automatic wait states follow. The interrupting device can gate up to a 16-bit identifier on to the bus when the \overline{DS} output is LOW. The WAIT input will be sampled again during the last TWA state. If the WAIT input is LOW one TW state will be inserted and the WAIT will be sampled during TW. Such TW insertion continues until the WAIT

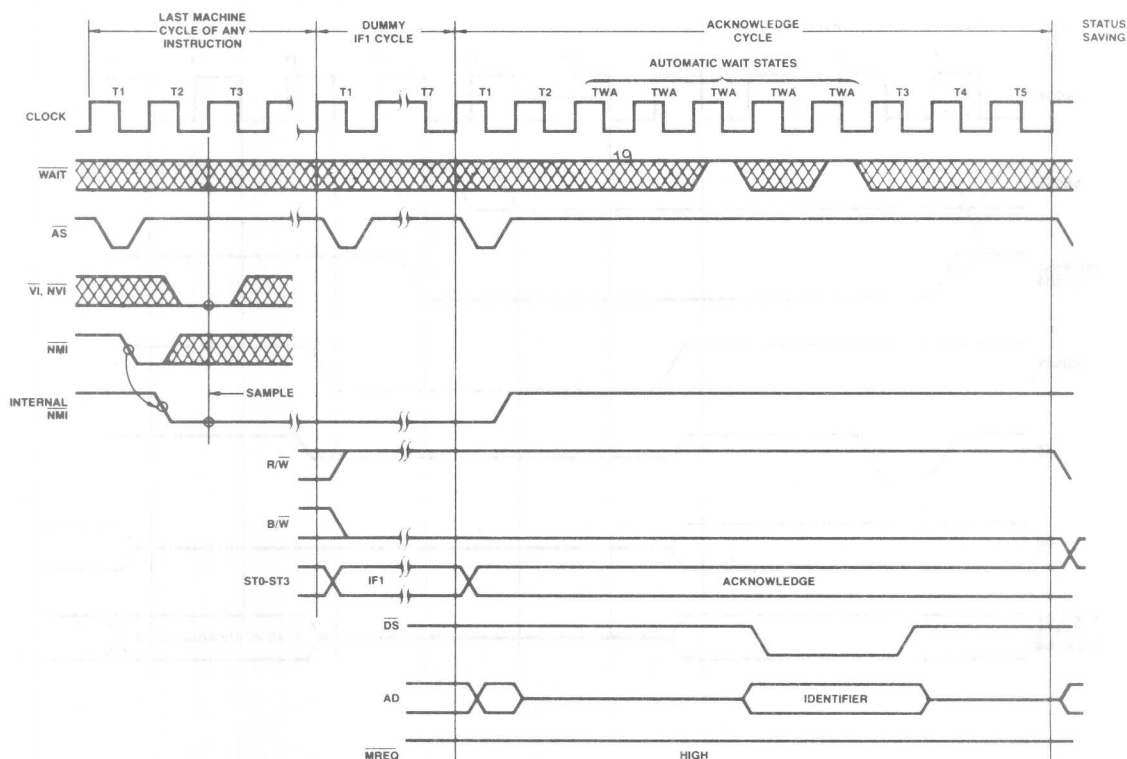


Figure 8. Interrupt Acknowledge Cycle.

MOS-234

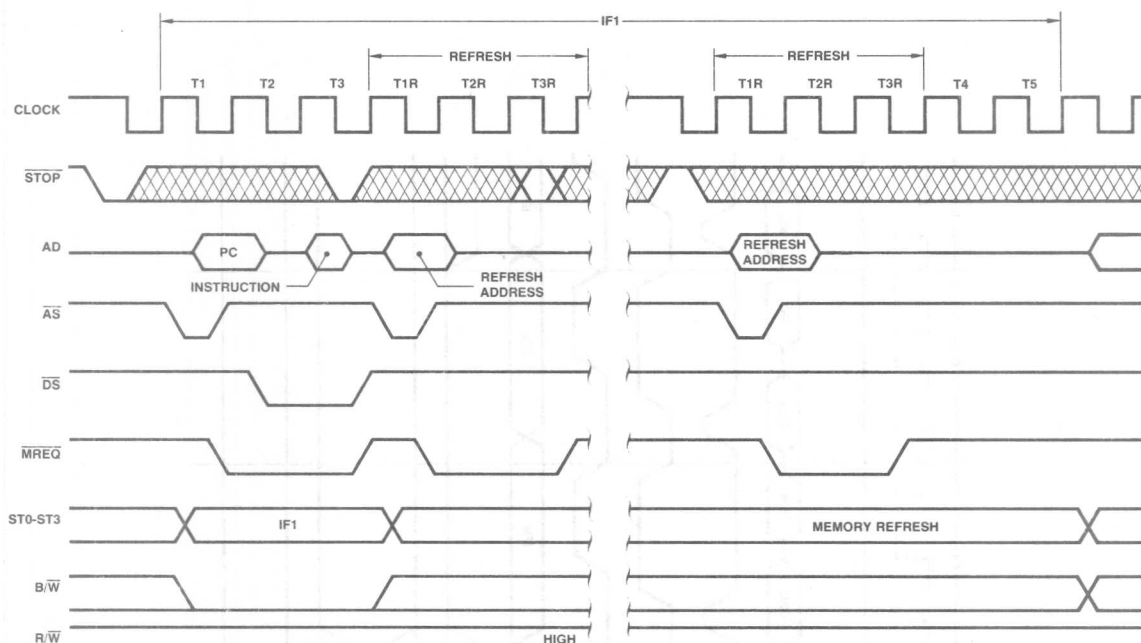


Figure 10. Single Step Timing.

MOS-236

MULTIMICROPROCESSOR FACILITIES

The AmZ8002 is provided with hardware and software facilities to support multiple microprocessor systems. The $\mu\bar{O}$ and $\mu\bar{I}$ signals of the AmZ8002 are used in conjunction with the MBIT, MREQ, MRES and MSET instructions for this purpose. The $\mu\bar{O}$ output can be activated LOW by using an appropriate instruction to signal a request from the AmZ8002 for a resource. The $\mu\bar{I}$ input is tested by the AmZ8002 before activating the $\mu\bar{O}$ output. LOW at the $\mu\bar{I}$ input at this time indicates that the resource is busy. The AmZ8002 can examine the $\mu\bar{I}$ input after activating the $\mu\bar{O}$ output LOW. The $\mu\bar{I}$ will be tested again to see if the requested resource became available. For detailed information on the Multimicroprocessor facilities the AmZ8001/AmZ8002 Processor Interface Manual (Publication No. AM-PUB089) should be consulted.

INITIALIZATION

A LOW on the $\overline{\text{Reset}}$ input starts the CPU initialization. The initialization sequence is shown in Figure 11. Within five clock periods after the HIGH to LOW level change of the $\overline{\text{Reset}}$ input the following will occur:

- AD0-AD15 bus will be in the HIGH impedance state
- $\overline{\text{AS}}$, $\overline{\text{DS}}$, $\overline{\text{MREQ}}$, $\overline{\text{BUSAK}}$ and $\mu\bar{O}$ outputs will be HIGH
- ST0-ST3 outputs will be LOW
- Refresh will be disabled
- $\text{R}/\overline{\text{W}}$, $\text{B}/\overline{\text{W}}$ and $\text{N}/\overline{\text{S}}$ outputs are not affected. For a power on reset the state of these outputs is not specified.

After the $\overline{\text{Reset}}$ input returns HIGH and remains HIGH for three clock periods, two 16-bit memory read operations will be performed as follows. Note that the $\text{N}/\overline{\text{S}}$ output will be LOW and ST0-ST3 outputs will reflect IFN code.

- The contents of the memory location 0002 will be read. This information will be loaded into the FCW of the AmZ8002.
- The contents of the memory location 0004 will be read. This information will be loaded into the AmZ8002 program counter.

This completes initialization sequence and an IF1 cycle will follow to fetch the first instruction to begin program execution. See the section on memory transactions for timing.

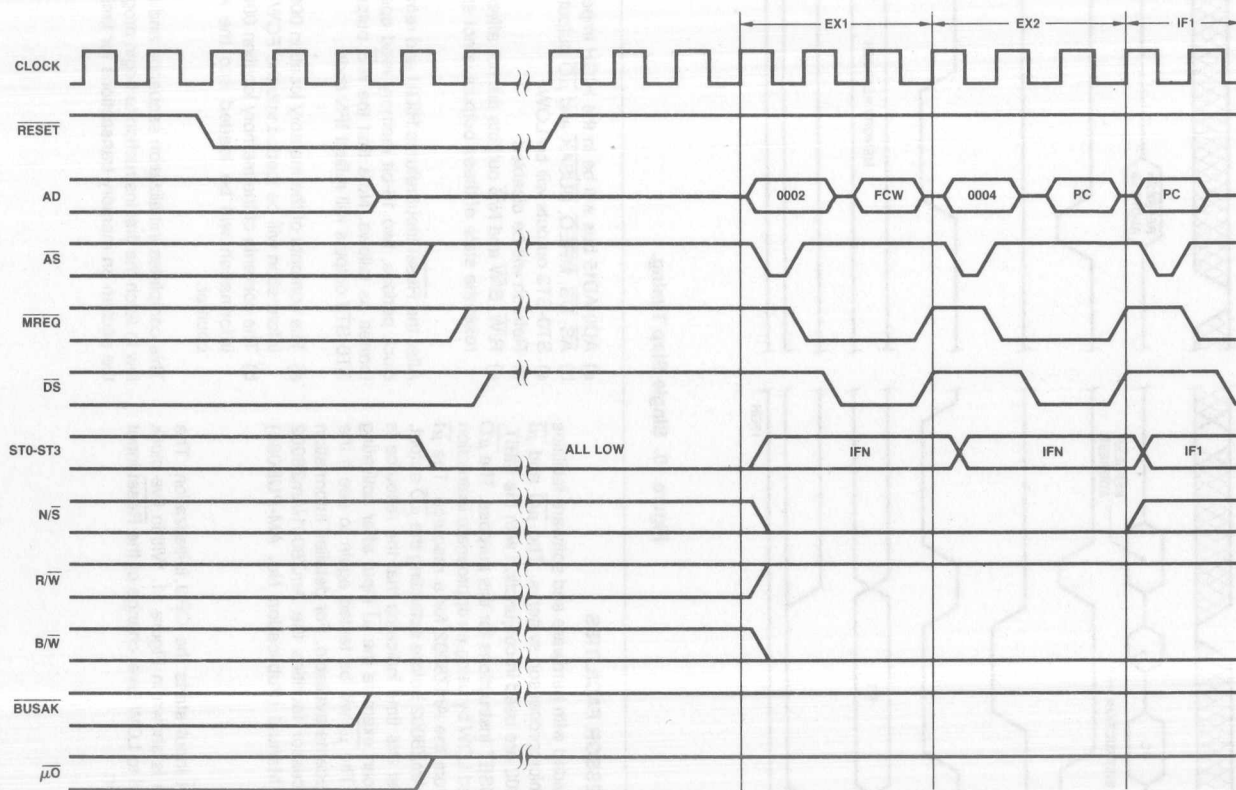


Figure 11. Reset Sequence.

AmZ8002 INSTRUCTION SET

LOAD AND EXCHANGE

Mne- monics	Operands	Addr. Modes	Operation
CLR CLRB	dst	R IR DA X	Clear $\text{dst} \leftarrow 0$
EX EXB	R, src	R IR DA X	Exchange $R \leftarrow \text{src}$
LD LDB LDL	R, src	R IM IM IR DA X BA BX	Load into Register $R \leftarrow \text{src}$
LD LDB LDL	dst, R	IR DA X BA BX	Load into Memory (Store) $\text{dst} \leftarrow R$
LD LDB	dst, IM	IR DA X	Load Immediate into Memory $\text{dst} \leftarrow \text{IM}$
LDA	R, src	DA X BA BX	Load Address $R \leftarrow \text{source address}$
LDAR	R, src	RA	Load Address Relative $R \leftarrow \text{source address}$
LDK	R, src	IM	Load Constant $R \leftarrow n$ ($n = 0 \dots 15$)
LDM	R, src, n	IR DA X	Load Multiple $R \leftarrow \text{src}$ (n consecutive words) ($n = 1 \dots 16$)
LDM	dst, R, n	IR DA X	Load Multiple (Store Multiple) $\text{dst} \leftarrow R$ (n consecutive words) ($n = 1 \dots 16$)
LDR LDRB LDRL	R, src	RA	Load Relative $R \leftarrow \text{src}$ (range $-32768 \dots +32767$)
LDR LDRB LDRL	dst, R	RA	Load Relative (Store Relative) $\text{dst} \leftarrow R$ (range $-32768 \dots +32767$)
POP POPL	dst, R	R IR DA X	Pop $\text{dst} \leftarrow \text{IR}$ Autoincrement contents of R
PUSH PUSHL	IR, src	R IM IR DA X	Push Autodecrement contents of R $\text{IR} \leftarrow \text{src}$

ARITHMETIC

Mne- monics	Operands	Addr. Modes	Operation
ADC ADCB	R, src	R	Add with Carry $R \leftarrow R + \text{src} + \text{carry}$
ADD ADDB ADDL	R, src	R IM IR DA X	Add $R \leftarrow R + \text{src}$
CP CPB CPL	R, src	R IM IR DA X	Compare with Register $R - \text{src}$
CP CPB	dst, IM	IR DA X	Compare with Immediate $\text{dst} - \text{IM}$
DAB	dst	R	Decimal Adjust
DEC DECB	dst, n	R IR DA X	Decrement by n $\text{dst} \leftarrow \text{dst} - n$ ($n = 1 \dots 16$)
DIV DIVL	R, src	R IM IR DA X	Divide (signed) Word: $R_{n+1} \leftarrow R_{n,n+1} \div \text{src}$ $R_n \leftarrow \text{remainder}$ Long Word: $R_{n+2,n+3} \leftarrow R_{n,n+3} \div \text{src}$ $R_{n,n+1} \leftarrow \text{remainder}$
EXTS EXTSB EXTSL	dst	R	Extend Sign Extend sign of low order half of st through high order half of dst
INC INCB	dst, n	R IR DA X	Increment by n $\text{dst} \leftarrow \text{dst} + n$ ($n = 1 \dots 16$)
MULT MULTL	R, src	R IM IR DA X	Multiply (signed) Word: $R_{n,n+1} \leftarrow R_{n+1} * \text{src}$ Long Word: $R_{n,n+3} \leftarrow R_{n+2,n+3} * \text{src}$ *Plus seven cycles for each 1 in the multiplicand
NEG NEGB	dst	R IR DA X	Negate $\text{dst} \leftarrow 0 - \text{dst}$
SBC SBCB	R, src	R	Subtract with Carry $R \leftarrow R - \text{src} - \text{carry}$
SUB SUBB SUBL	R,src	R IM IR DA X	Subtract $R \leftarrow R - \text{src}$

LOGICAL

Mne- monics	Operands	Addr. Modes	Operation
AND ANDB	R, src	R IM IR DA X	AND $R \leftarrow R \text{ AND } \text{src}$
COM COMB	dst	R IM IR DA X	Complement $\text{dst} \leftarrow \text{NOT } \text{dst}$
OR ORB	R, src	R IM IR DA X	OR $R \leftarrow R \text{ OR } \text{src}$
TEST TESTB TESTL	dst	R IR DA X	TEST $\text{dst OR } 0$
TCC TCCB	cc, dst	R	Test Condition Code Set LSB if cc is true
XOR XORB	R, src	R IM IR DA X	Exclusive OR $R \leftarrow R \text{ XOR } \text{src}$

PROGRAM CONTROL

Mne- monics	Operands	Addr. Modes	Operation
CALL	dst	IR DA X	Call Subroutine Autodecrement SP $@ \text{SP} \leftarrow \text{PC}$ $\text{PC} \leftarrow \text{dst}$
CALR	dst	RA	Call Relative Autodecrement SP $@ \text{SP} \leftarrow \text{PC}$ $\text{PC} \leftarrow \text{PC} + \text{dst}$ (range -4096 to +4096)
DJNZ DBJNZ	R, dst	RA	Decrement and Jump if Non-Zero $R \leftarrow R - 1$ If $R = 0$: $\text{PC} \leftarrow \text{PC} + \text{dst}$ (range -254 to 0)
IRET*	—	—	Interrupt Return $\text{PS} \leftarrow @ \text{SP}$ Autoincrement SP
JP	cc, dst	IR IR DA X	Jump Conditional If cc is true: $\text{PC} \leftarrow \text{dst}$
JR	cc, dst	RA	Jump Conditional Relative If cc is true: $\text{PC} \leftarrow \text{PC} + \text{dst}$ (range -256 to +254)
RET	cc	—	Return Conditional If cc is true: $\text{PC} \leftarrow @ \text{SP}$ Autodecrement SP
SC	src	IM	System Call Autodecrement SP $@ \text{SP} \leftarrow \text{old PS}$ Push instruction $\text{PS} \leftarrow \text{System Call PS}$

*Privileged instructions. Executed in system mode only.

BIT MANIPULATION

Mne- monics	Operand	Addr. Modes	Operation
BIT BITB	dst, b	R IR DA X	Test Bit Static $Z \text{ flag} \leftarrow \text{NOT } \text{dst bit specified by } b$
BIT BITB	dst, R	R	Test Bit Dynamic $Z \text{ flag} \leftarrow \text{NOT } \text{dst bit specified by contents of } R$
RES RESB	dst, b	R IR DA X	Reset Bit Static Reset dst bit specified by b
RES RESB	dst, R	R	Reset Bit Dynamic Reset dst bit specified by contents of R
SET SETB	dst, b	R IR DA X	Set Bit Static Set dst bit specified by b
SET SETB	dst, R	R	Set Bit Dynamic Set dst bit specified by contents of R
TSET TSETB	dst	R IR DA X	Test and Set $S \text{ flag} \leftarrow \text{MSB of } \text{dst}$ $\text{dst} \leftarrow \text{all } 1\text{s}$

ROTATE AND SHIFT

Mne- monics	Operand	Addr. Modes	Operation
RLDB	R, src	R	Rotate Digit Left
RRDB	R, src	R	Rotate Digit Right
RL RLB	dst, n	R R	Rotate Left by n bits ($n = 1, 2$)
RLC RLCB	dst, n	R R	Rotate Left through Carry by n bits ($n = 1, 2$)
RR RRB	dst, n	R R	Rotate Right by n bits ($n = 1, 2$)
RRC RRCB	dst, n	R R	Rotate Right through Carry by n bits ($n = 1, 2$)
SDA SDAB SDAL	dst, R	R	Shift Dynamic Arithmetic Shift dst left or right by contents of R
SDL SDLB SDLL	dst, R	R	Shift Dynamic Logical Shift dst left or right by contents of R
SLA SLAB SLAL	dst, n	R	Shift Left Arithmetic by n bits
SLL SLLB SLLL	dst, n	R	Shift Left Logical by n bits
SRA SRAB SRAL	dst, n	R	Shift Right Arithmetic by n bits
SRL SRLB SRLl	dst, n	R	Shift Right Logical by n bits

BLOCK TRANSFER AND STRING MANIPULATION

Mne- monics	Operands	Addr. Modes	Operation
CPD CPDB	R_X , src, R_Y , cc	IR	Compare and Decrement $R_X \leftarrow \text{src}$ Autodecrement src address $R_Y \leftarrow R_Y - 1$
CPDR CPDRB	R_X , src, R_Y , cc	IR	Compare, Decrement and Repeat $R_X \leftarrow \text{src}$ Autodecrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
CPI CPIB	R_X , src, R_Y , cc	IR	Compare and Increment $R_X \leftarrow \text{src}$ Autoincrement src address $R_Y \leftarrow R_Y + 1$
CPIR CPIRB	R_X , src, R_Y , cc	IR	Compare, Increment and Repeat $R_X \leftarrow \text{src}$ Autoincrement src address $R_Y \leftarrow R_Y + 1$ Repeat until cc is true or $R_Y = 0$
CPSD CPSDB	dst, src, R , cc	IR	Compare String and Decrement dst \leftarrow src Autodecrement dst and src addresses $R \leftarrow R - 1$
CPSDR CPSDRB	dst, src, R , cc	IR	Compare String, Decr. and Repeat dst \leftarrow src Autodecrement dst and src addresses $R \leftarrow R - 1$ Repeat until cc is true or $R = 0$
CPSI CPSIB	dst, src, R , cc	IR	Compare String and Increment dst \leftarrow src Autoincrement dst and src addresses $R \leftarrow R + 1$
CPSIR CPSIRB	dst, src, R , cc	IR	Compare String, Incr. and Repeat dst \leftarrow src Autoincrement dst and src addresses $R \leftarrow R + 1$ Repeat until cc is true or $R = 0$
LDD Lddb	dst, src, R	IR	Load and Decrement dst \leftarrow src Autodecrement dst and src addresses $R \leftarrow R - 1$
LDDR LDRB	dst, src, R	IR	Load, Decrement and Repeat dst \leftarrow src Autodecrement dst and src addresses $R \leftarrow R - 1$ Repeat until $R = 0$

BLOCK TRANSFER AND STRING MANIPULATION (Cont.)

Mne- monics	Operands	Addr. Modes	Operation
LDI LDIB	dst, src, R	IR	Load and Increment dst \leftarrow src Autoincrement dst and src addresses $R \leftarrow R + 1$
LDIR LDIRB	dst, src, R	IR	Load, Increment and Repeat dst \leftarrow src Autoincrement dst and src addresses $R \leftarrow R + 1$ Repeat until $R = 0$
TRDB	dst, src, R	IR	Translate and Decrement dst \leftarrow src (dst) Autodecrement dst address $R \leftarrow R - 1$
TRDRB	dst, src, R	IR	Translate, Decrement and Repeat dst \leftarrow src (dst) Autodecrement dst address $R \leftarrow R - 1$ Repeat until $R = 0$
TRIB	dst, src, R	IR	Translate and Increment dst \leftarrow src (dst) Autoincrement dst address $R \leftarrow R + 1$
TRIRB	dst, src, R	IR	Translate, Increment and Repeat dst \leftarrow src (dst) Autoincrement dst address $R \leftarrow R + 1$ Repeat until $R = 0$
TRTDB	src 1, src 2, R	IR	Translate and Test, Decrement $RH1 \leftarrow \text{src 2 (src 1)}$ Autodecrement src 1 address $R \leftarrow R - 1$
TRTDRB	src 1, src 2, R	IR	Translate and Test, Decrement and Repeat $RH1 \leftarrow \text{src 2 (src 1)}$ Autodecrement src 1 address $R \leftarrow R - 1$ Repeat until $R = 0$ or $RH1 = 0$
TRTIB	src 1, src 2, R	IR	Translate and Test, Increment $RH1 \leftarrow \text{src 2 (src 1)}$ Autoincrement src 1 address $R \leftarrow R + 1$
TRTIRB	src 1, src 2, R	IR	Translate and Test, Increment and Repeat $RH1 \leftarrow \text{src 2 (src 1)}$ Autoincrement src 1 address $R \leftarrow R + 1$ Repeat until $R = 0$ or $RH1 = 0$

INPUT/OUTPUT

Mne- monics	Operands	Addr. Modes	Operation
IN* INB*	R, src	IR DA	Input $R \leftarrow \text{src}$
IND* INDB*	dst, src, R	IR	Input and Decrement $\text{dst} \leftarrow \text{src}$ Autodecrement dst address $R \leftarrow R - 1$
INDR* INDRB*	dst, src, R	IR	Input, Decrement and Repeat $\text{dst} \leftarrow \text{src}$ Autodecrement dst address $R \leftarrow R - 1$ Repeat until $R = 0$
INI* INIB*	dst, src, R	IR	Input and Increment $\text{dst} \leftarrow \text{src}$ Autoincrement dst address $R \leftarrow R + 1$
INIR* INIRB*	dst, src, R	IR	Input, Increment and Repeat $\text{dst} \leftarrow \text{src}$ Autoincrement dst address $R \leftarrow R + 1$ Repeat until $R = 0$
OUT* OUTB*	dst, R	IR DA	Output $\text{dst} \leftarrow R$
OUTD* OUTDB*	dst, src, R	IR	Output and Decrement $\text{dst} \leftarrow \text{src}$ Autodecrement src address $R \leftarrow R - 1$
OTDR* OTDRB*	dst, src, R	IR	Output, Decrement and Repeat $\text{dst} \leftarrow \text{src}$ Autodecrement src address $R \leftarrow R - 1$ Repeat until $R = 0$
OUTI* OUTIB*	dst, src, R	IR	Output and Increment $\text{dst} \leftarrow \text{src}$ Autoincrement src address $R \leftarrow R + 1$
OTIR* OTIRB*	dst, src, R	IR	Output, Increment and Repeat $\text{dst} \leftarrow \text{src}$ Autoincrement src address $R \leftarrow R + 1$ Repeat until $R = 0$
SIN* SINB*	R, src	DA	Special Input $R \leftarrow \text{src}$
SIND* SINDB*	dst, src, R	IR	Special Input and Decrement $\text{dst} \leftarrow \text{src}$ Autodecrement dst address $R \leftarrow R - 1$
SINDR* SINDRB*	dst, src, R	IR	Special Input, Decr. and Repeat $\text{dst} \leftarrow \text{src}$ Autodecrement dst address $R \leftarrow R - 1$ Repeat until $R = 0$
SINI* SINIB*	dst, src, R	IR	Special Input and Increment $\text{dst} \leftarrow \text{src}$ Autoincrement dst address $R \leftarrow R + 1$
SINIR* SINIRB*	dst, src, R	IR	Special Input, Incr. and Repeat $\text{dst} \leftarrow \text{src}$ Autoincrement dst address $R \leftarrow R + 1$ Repeat until $R = 0$

INPUT/OUTPUT (Cont.)

Mne- monics	Operands	Addr. Modes	Operation
SOUT* SOUTB*	dst, src	DA	Special Output $\text{dst} \leftarrow \text{src}$
SOUTD* SOUTDB*	dst, src, R	IR	Special Output and Decrement $\text{dst} \leftarrow \text{src}$ Autodecrement src address $R \leftarrow R - 1$
SOTDR* SOTDRB*	dst, src, R	IR	Special Output, Decr. and Repeat $\text{dst} \leftarrow \text{src}$ Autodecrement src address $R \leftarrow R - 1$ Repeat until $R = 0$
SOUTI* SOUTIB*	dst, src, R	IR	Special Output and Increment $\text{dst} \leftarrow \text{src}$ Autoincrement src address $R \leftarrow R + 1$
SOTIR* SOTIRB*	dst, src, R	R	Special Output, Incr. and Repeat $\text{dst} \leftarrow \text{src}$ Autoincrement src address $R \leftarrow R + 1$ Repeat until $R = 0$

CPU CONTROL

Mne- monics	Operands	Addr. Modes	Operation
COMFLG	flags	—	Complement Flag (Any combination of C, Z, S, P/V)
DI*	int	—	Disable Interrupt (Any combination of NVI, VI)
EI*	int	—	Enable Interrupt (Any combination of NVI, VI)
HALT*	—	—	HALT
LDCTL*	CTLR, src	R	Load into Control Register $\text{CTLR} \leftarrow \text{src}$
LDCTL*	dst, CTLR	R	Load from Control Register $\text{dst} \leftarrow \text{CTLR}$
LDCTLB	FLGR, src	R	Load into Flag Byte Register $\text{FLGR} \leftarrow \text{src}$
LDCTLB	dst, FLGR	R	Load from Flag Byte Register $\text{dst} \leftarrow \text{FLGR}$
LDPS*	src	IR DA X	Load Program Status $\text{PS} \leftarrow \text{src}$
MBIT*	—	—	Test Multi-Micro Bit Set S if $\overline{\mu I}$ is High; reset S if $\overline{\mu I}$ is Low.
MREQ*	dst	R	Multi-Micro Request
MRES*	—	—	Multi-Micro Reset
MSET*	—	—	Multi-Micro Set
NOP	—	—	No Operation
RESFLG	flag	—	Reset Flag (Any combination of C, Z, S, P/V)
SETFLG	flag	—	Set Flag (Any combination of C, Z, S, P/V)

*Privileged instructions. Executed in system mode only.

MAXIMUM RATINGS above which useful life may be impaired

Voltages on all inputs and outputs with respect to GND	-0.3 to +7.0V
Ambient Temperature under bias	0 to 70°C
Storage Temperature	-65 to +150°C

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

ELECTRICAL CHARACTERISTICS over operating range (Note 1)**AmZ8002DC**

Parameter	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	$V_{CC} - 0.4$	$V_{CC} + 0.3$	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	-0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	$V_{CC} + 0.3$	Volts
V_{IL}	Input Low Voltage		-0.3	0.8	Volts
V_{OH}	Output High Voltage	$I_{OH} = -250\mu A$	2.4		Volts
V_{OL}	Output Low Voltage	$I_{OL} = +2.0mA$		0.4	Volts
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4V$		± 10	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		± 10	μA
I_{CC}	V_{CC} Supply Current			300	mA

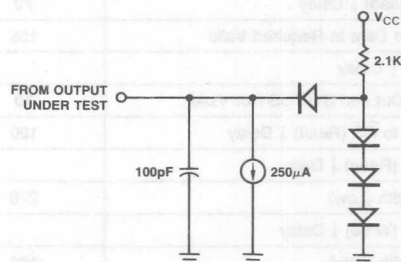
Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

$$+4.75V \leq V_{CC} \leq +5.25V$$

$$GND = 0V$$

$$0^\circ C \leq T_A \leq +70^\circ C$$



MOS-238

All AC parameters assume a load capacitance of 100pF max. Timing references between two output signals assume a load difference of 50pF max.

AmZ8002

SWITCHING CHARACTERISTICS over operating range

AmZ8002DC

Number	Parameter	Description	Min	Max	Units
1	TcC	Clock Cycle Time	250	2000	ns
2	TwCh	Clock Width (High)	105	2000	ns
3	TwCl	Clock Width (Low)	105	2000	ns
4	TfC	Clock Fall Time		20	ns
5	TrC	Clock Rise Time		20	ns
6					
7					
8	TdC(Bz)	Clock \uparrow to Bus Float		65	ns
9	TdC(A)	Clock \uparrow to Address Valid		100	ns
10	TdC(Az)	Clock \uparrow to Address Float		65	ns
11	TdA(DI)	Address Valid to Data In Required Valid	400		ns
12	TsDI(C)	Data In to Clock \downarrow Set-up Time	70		ns
13	TdDS(A)	$\overline{DS} \uparrow$ to Address Active	80		ns
14	TdC(DO)	Clock \uparrow to Data Out Valid		100	ns
15	ThDI(DS)	Data In to $\overline{DS} \uparrow$ Hold Time	0		ns
16	TdDO(DS)	Data Out Valid to $\overline{DS} \uparrow$ Delay	230		ns
17	TdA(MR)	Address Valid to $\overline{MREQ} \downarrow$ Delay	55		ns
18	TdC(MR)	Clock \downarrow to $\overline{MREQ} \downarrow$ Delay		80	ns
19	TwMRh	\overline{MREQ} Width (High)	190		ns
20	TdMR(A)	$\overline{MREQ} \downarrow$ to Address Not Active	70		ns
21	TdDO(DSW)	Data Out Valid to $\overline{DS} \downarrow$ (Write) Delay	55		ns
22	TdMR(DI)	$\overline{MREQ} \downarrow$ to Data In Required Valid	330		ns
23	TdC(MR)	Clock \downarrow to $\overline{MREQ} \uparrow$ Delay		80	ns
24	TdC(ASf)	Clock \uparrow to $\overline{AS} \downarrow$ Delay		80	ns
25	TdA(AS)	Address Valid to $\overline{AS} \uparrow$ Delay	55		ns
26	TdC(ASr)	Clock \downarrow to $\overline{AS} \uparrow$ Delay		90	ns
27	TdAS(DI)	$\overline{AS} \uparrow$ to Data In Required Valid	290		ns
28	TdDS(AS)	$\overline{DS} \uparrow$ to $\overline{AS} \downarrow$ Delay	70		ns
29	TwAS	\overline{AS} Width (Low)	80		ns
30	TdAS(A)	$\overline{AS} \uparrow$ to Address Not Active Delay	60		ns
31	TdAz(DSR)	Address Float to \overline{DS} (Read) \downarrow Delay	0		ns
32	TdAS(DSR)	$\overline{AS} \uparrow$ to \overline{DS} (Read) \downarrow Delay	70		ns
33	TdDSR(DI)	\overline{DS} (Read) \downarrow to Data In Required Valid	155		ns
34	TdC(DSr)	Clock \downarrow to $\overline{DS} \uparrow$ Delay		70	ns
35	TdDS(DO)	$\overline{DS} \uparrow$ to Data Out and STATUS Not Valid	80		ns
36	TdA(DSR)	Address Valid to \overline{DS} (Read) \downarrow Delay	120		ns
37	TdC(DSR)	Clock \uparrow to \overline{DS} (Read) \downarrow Delay		120	ns
38	TwDSR	\overline{DS} (Read) Width (Low)	275		ns
39	TdC(DSW)	Clock \downarrow to \overline{DS} (Write) \downarrow Delay		95	ns
40	TwDSW	\overline{DS} (Write) Width (Low)	160		ns
41	TdDSI(DI)	\overline{DS} (Input) \downarrow to Data In Required Valid	315		ns
42	TdC(DSf)	Clock \downarrow to \overline{DS} (I/O) \downarrow Delay		120	ns
43	TwDS	\overline{DS} (I/O) Width (Low)	400		ns
44	TdAS(DSA)	$\overline{AS} \uparrow$ to \overline{DS} (Acknowledge) \downarrow Delay	960		ns
45	TdC(DSA)	Clock \uparrow to \overline{DS} (Acknowledge) \downarrow Delay		120	ns
46	TdDSA(DI)	\overline{DS} (Acknowledge) \downarrow to Data In Required Delay	420		ns
47	TdC(S)	Clock \uparrow to Status Valid Delay		110	ns
48	TdS(AS)	Status Valid to $\overline{AS} \uparrow$ Delay	40		ns

SWITCHING CHARACTERISTICS (Cont.)

AmZ8002DC

Number	Parameter	Description	Min	Max	Units
49	TsR(C)	$\overline{\text{RESET}}$ to Clock \uparrow Set-up Time	180		ns
50	ThR(C)	$\overline{\text{RESET}}$ to Clock \uparrow Hold Time	0		ns
51	TwNMI	NMI Width (Low)	100		ns
52	TsNMI(C)	NMI to Clock \uparrow Set-up Time	140		ns
53	TsVI(C)	$\overline{\text{VI}}$, $\overline{\text{NVI}}$ to Clock \uparrow Set-up Time	110		ns
54	ThVI(C)	$\overline{\text{VI}}$, $\overline{\text{NVI}}$ to Clock \uparrow Hold Time	0		ns
55					
56					
57	Ts μ i(C)	$\overline{\mu\text{i}}$ to Clock \uparrow Set-up Time	180		ns
58	Th μ i(C)	$\overline{\mu\text{i}}$ to Clock \uparrow Hold Time	0		ns
59	TdC(μ o)	Clock \uparrow to $\overline{\mu\text{o}}$ Delay		120	ns
60	TsSTP(C)	$\overline{\text{STOP}}$ to Clock \downarrow Set-up Time	140		ns
61	ThSTP(C)	$\overline{\text{STOP}}$ to Clock \downarrow Hold Time	0		ns
62	TsWT(C)	$\overline{\text{WAIT}}$ to Clock \downarrow Set-up Time	70		ns
63	ThWT(C)	$\overline{\text{WAIT}}$ to Clock \downarrow Hold Time	0		ns
64	TsBRQ(C)	$\overline{\text{BUSRQ}}$ to Clock \uparrow Set-up Time	90		ns
65	ThBRQ(C)	$\overline{\text{BUSRQ}}$ to Clock \uparrow Hold Time	0		ns
66	TdC(BAKr)	Clock \uparrow to $\overline{\text{BUSAk}}$ \uparrow Delay		100	ns
67	TdC(BAKf)	Clock \uparrow to $\overline{\text{BUSAk}}$ \downarrow Delay		100	ns

For more information, refer to these AMD publications:

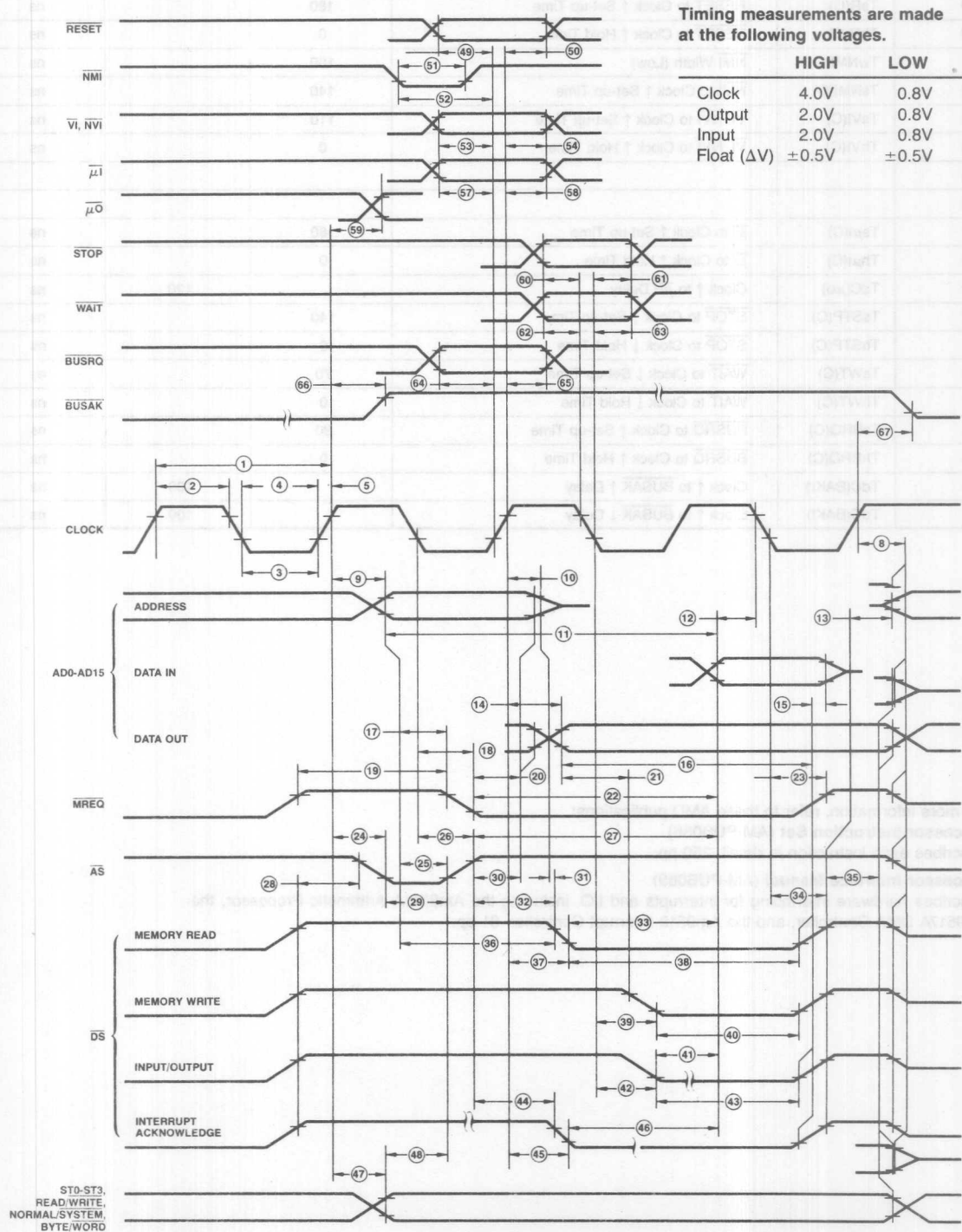
Processor Instruction Set (AM-PUB086).

Describes each instruction in detail. 250 pp.

Processor Interface Manual (AM-PUB089).

Describes hardware interfacing for interrupts and I/O, including the Am9511A Arithmetic Processor, the Am9517A DMA Controller, and the Am9519 Interrupt Controller. 81 pp.

AC TIMING DIAGRAM



This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

MOS-239

I/O, including the Am9511A Arithmetic Processor, the
nt Controller, 81 pp.

I/O, including the Am9511A Arithmetic Processor, the

pt Controller. 81 pp.

Am8085A/Am8085A-2/Am9085ADM

Single Chip 8-Bit N-Channel Microprocessor

DISTINCTIVE CHARACTERISTICS

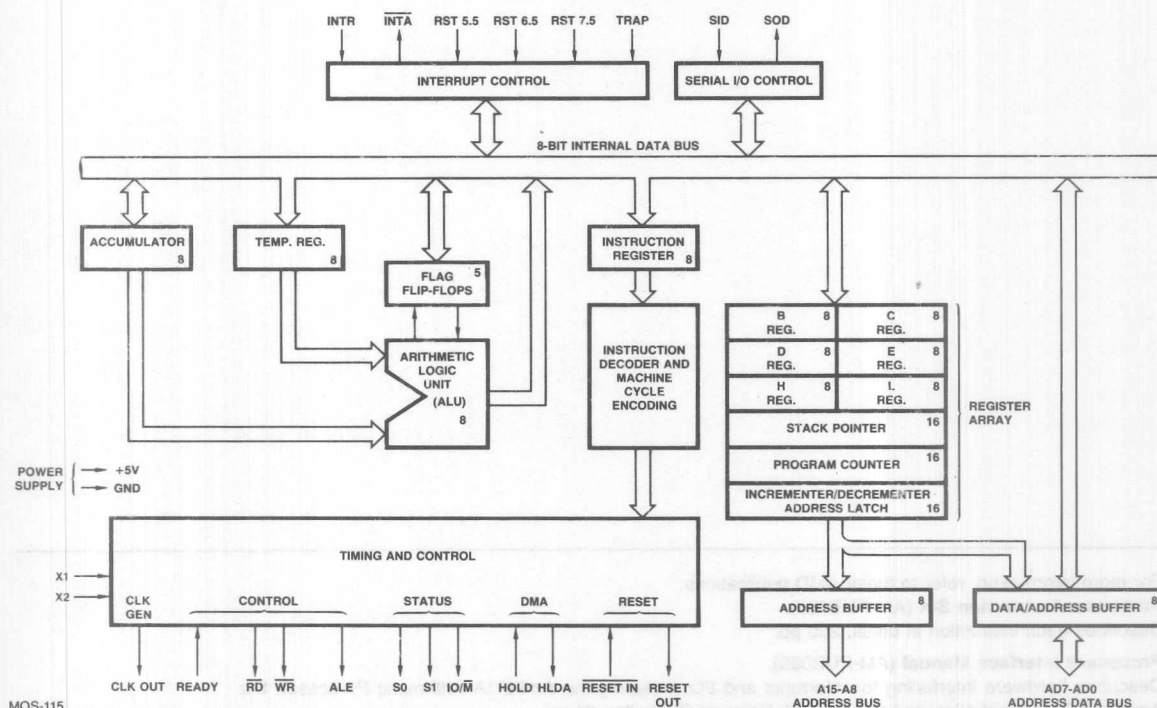
- Complete 8-bit parallel CPU
- On-chip system controller; advanced cycle status information available for large system control
- Four vectored interrupts (one is non-maskable)
- On-chip clock generator (with external crystal, LC or R/C network)
- Serial in/serial out port
- Decimal, binary and double precision arithmetic
- Direct addressing capability to 64K bytes of memory
- 1.3μs instruction cycle (Am8085A)
- 0.8μs instruction cycle (Am8085A-2)
- 100% software compatible with Am9080A
- Single +5V power supply
- 100% MIL-STD-883, Level C processing

GENERAL DESCRIPTION

The Am8085A is a new generation, complete 8-bit parallel central processing unit (CPU). Its instruction set is 100% software compatible with the Am9080A microprocessor. Specifically, the Am8085A incorporates all of the features that the Am8224 (clock generator) and Am8228 (system controller) provided for the Am9080A. The Am8085A-2 is a faster version of the Am8085A.

The Am8085A uses a multiplexed Data Bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of Am8155/Am8355 memory products allows a direct interface with Am8085A. The Am8085A components, including various timing compatible support chips, allow system speed optimization.

BLOCK DIAGRAM



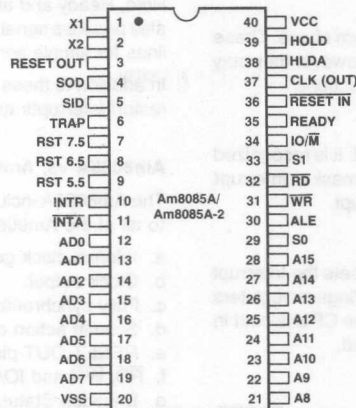
ORDERING INFORMATION

Package Type	Temperature Range	Maximum Clock Frequency	
		3MHz	5MHz
Molded DIP	$0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$	AM8085APC/P8085A	AM8085A-2PC/P8085A-2
Hermetic DIP*		AM8085ADC/D8085A	AM8085A-2DC/D8085A-2
		AM8085ACC/C8085A	AM8085A-2CC/C8085A-2
Hermetic DIP	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$	AM9085ADM	

*Hermetic = Ceramic = DC = CC = D-40-1.

CONNECTION DIAGRAM

Top View



Note: Pin 1 is marked for orientation.

Figure 1.

MOS-116

Am8085A FUNCTIONAL PIN DEFINITION

The following describes the function of each pin:

A8-A15 (Output 3-State)

Address Bus — the most significant eight bits of the memory address or the eight bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.

AD0-AD7 (Input/Output 3-State)

Multiplexed Address/Data Bus — lower eight bits of the memory address (or I/O address) appear on the bus during the first clock cycle of a machine cycle. It then becomes the data bus during the second and third clock cycles.

Three-stated during Hold and Halt modes.

ALE (Output)

Address Latch Enable — it occurs during the first clock cycle of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. The falling edge ALE can also be used to strobe the status information. ALE is never 3-stated.

S0, S1 (Output)

Data Bus Status. Encoded status of the bus cycle.

S1	S0	
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

S1 can be used as an advanced R/W status.

 \overline{RD} (Output 3-State)

READ — A low level on \overline{RD} indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer. Three-stated during Hold and Halt and during RESET.

 \overline{WR} (Output 3-State)

WRITE — A low level on \overline{WR} indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of \overline{WR} . Three-stated during Hold and Halt modes.

READY (Input)

If READY is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is low, the CPU will wait an integral number of clock cycles for READY to go high before completing the read or write cycle.

HOLD (Input)

HOLD — indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request, will relinquish the use of buses as soon as the completion of the current machine cycle. Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, RD, WR and IO/M lines are three-stated.

HLDA (Output)

HOLD ACKNOWLEDGE — indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle. HLDA goes low after the Hold request is removed. The CPU takes the buses one half clock cycle after HLDA goes LOW.

INTR (Input)

INTERRUPT REQUEST — is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an \overline{INTA} will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

 \overline{INTA} (Output)

INTERRUPT ACKNOWLEDGE — is used instead of (and has the same timing as) RD during the Instruction cycle after an INTR is accepted. It can be used to activate the Am9519 Interrupt chip or some other interrupt port.

RST 5.5	} (Inputs)
RST 6.5	
RST 7.5	

RESTART INTERRUPTS — these three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted.

RST 7.5 → Highest Priority
 RST 6.5
 RST 5.5 → Lowest Priority

The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR. However they may be individually masked out using the SIM instructions.

TRAP (Input)

Trap interrupt is a non-maskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt.

RESET IN (Input)

Reset sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. None of the other flags or registers (except the instruction register) are affected. The CPU is held in the reset condition as long as RESET is applied.

RESET OUT (Output)

Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.

X1, X2 (Input)

Crystal, LC or R/C network connections to set the internal clock generator. X1 can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.

CLK (Output)

Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X1, X2 input period.

IO/M (Output)

IO/M indicates whether the Read/Write is to memory or I/O. 3-stated during Hold and Halt modes.

SID (Input)

Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

SOD (Output)

Serial output data line. The output SOD is set or reset as specified by the SIM instruction.

VCC

+5 volt supply.

VSS

Ground reference.

FUNCTIONAL DESCRIPTION

The Am8085A is a complete 8-bit parallel central processor. It is designed with N-channel depletion loads and requires a single +5 volt supply. Its basic clock speed is 3MHz (5MHz: Am8085A-2) thus improving on the present Am9080's performance with higher system speed. Also it is designed to fit into a minimum system of three ICs: The CPU, a RAM/IO, and a ROM or PROM/IO chip.

The Am8085A uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first cycle the address is sent out. The lower eight bits are latched into the peripherals by the Address Latch Enable (ALE). During the rest of the machine cycle the Data Bus is used for memory or I/O data.

The Am8085A provides \overline{RD} , \overline{WR} and $\overline{IO}/\overline{Memory}$ signals for bus control. An Interrupt Acknowledge signal (INTA) is also provided. Hold, Ready and all Interrupts are synchronized. The Am8085A also provides serial input data (SID) and serial output data (SOD) lines for simple serial interface.

In addition to these features, the Am8085A has three maskable, restart interrupts and one non-maskable trap interrupt.

Am8085A vs. Am8080A

The Am8085A includes the following features on-chip in addition to all of the Am9080A functions.

- Internal clock generator
- Clock output
- Fully synchronized Ready
- Schmitt action on RESET IN
- RESET OUT pin
- \overline{RD} , \overline{WR} and $\overline{IO}/\overline{M}$ Bus Control Signals
- Encoded Status information
- Multiplexed Address and Data
- Direct Restarts and non-maskable Interrupt
- Serial Input/Output lines

The internal clock generator requires an external crystal or R/C network. It will oscillate at twice the basic CPU operating frequency. A 50% duty cycle, two phase, non-overlapping clock is generated from this oscillator internally and one phase of the clock ($\phi 2$) is available as an external clock. The Am8085A directly provides the external RDY synchronization previously provided by the Am8224. The RESET IN input is provided with a Schmitt action input so that power-on reset only requires a resistor and capacitor. RESET OUT is provided for System RESET.

The Am8085A provides \overline{RD} , \overline{WR} and $\overline{IO}/\overline{M}$ signals for Bus control. An INTA which was previously provided by the Am8228 in Am9080A systems is also included in Am8085A.

STATUS INFORMATION

Status information is directly available from the Am8085A. ALE serves as a status strobe. The status is partially encoded and provides the user with advanced timing of the type of bus transfer being done. IO/M cycle status signal is provided directly also. Decoded S0, S1 carries the following status information:

MACHINE CYCLE STATUS

IO/M	S1	S0	Status
0	0	1	Memory write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch
1	1	1	Interrupt Acknowledge
.	0	0	Halt
.	X	X	Hold
.	X	X	Reset

. = 3-state (high impedance)

X = unspecified

S1 can be interpreted as $\overline{R}/\overline{W}$ in all bus transfers.

In the Am8085A the eight LSB of address are multiplexed with the data instead of status. The ALE line is used as a strobe to enter the lower half of the address into the memory or peripheral address latch. This also frees extra pins for expanded interrupt capability.

INTERRUPT AND SERIAL I/O

The Am8085A/Am8085A-2 has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP. INTR is identical in function to the Am8080A INT. Each of three RESTART inputs, 5.5, 6.5, 7.5, has programmable mask. TRAP is also a RESTART interrupt except it is non-maskable.

The three RESTART interrupts cause the internal execution of RST (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes the internal execution of a RST independent of the state of the interrupt enable or masks.

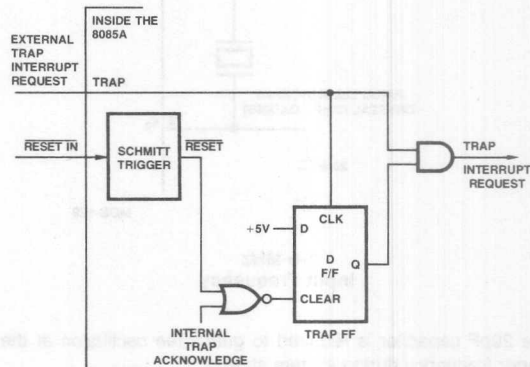
Name	RESTART Address (Hex)
TRAP	24 ₁₆
RST 5.5	2C ₁₆
RST 6.5	34 ₁₆
RST 7.5	3C ₁₆

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are high level-sensitive like INTR (and INT on the Am8080A) and are recognized with the same timing as INTR. RST 7.5 is rising edge-sensitive. For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request. The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a RESET IN to the Am8085A. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The status of the three RST interrupt masks can only be affected by the SIM instruction and RESET IN.

The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending as follows: TRAP — highest priority, RST 7.5, RST 6.5, RST 5.5, INTR — lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt a RST 7.5 routine if the interrupts were re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic errors such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both edge and level sensitive. The TRAP input must go high and remain high to be acknowledged, but will not be recognized again until it goes low, then high again. This avoids any false triggering due to noise or logic glitches. The following diagram illustrates the TRAP interrupt request circuitry within the Am8085A.



MOS-117

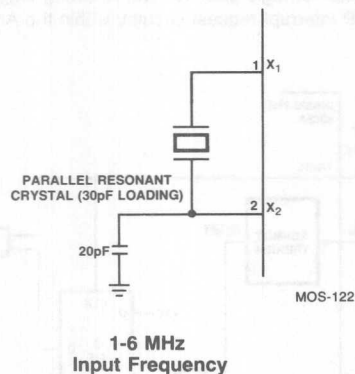
Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

The TRAP interrupt is special in that it preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status.

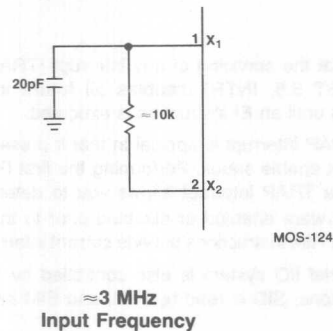
The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD data.

DRIVING THE X1 and X2 INPUTS

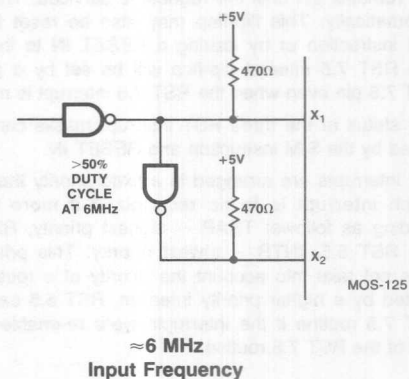
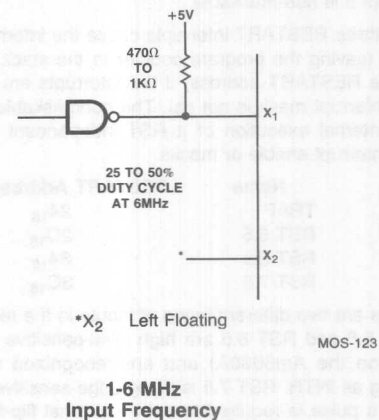
The user may drive the X1 and X2 inputs of the Am8085A or Am8085A-2 with a crystal, an external clock source or an R/C network as shown below. The driving frequency must be twice the desired internal operating frequency (the Am8085A would require a 6MHz crystal for 3MHz internal operation).



The 20pF capacitor is required to guarantee oscillation at the proper frequency during system startup.



RC Mode causes a large drift in clock frequency because of the variation in on-chip timing generation parameters. Use of RC Mode should be limited to an application which can tolerate a wide frequency variation.



Note: Duty cycle refers to the percentage of the clock input cycle when X₁ is high.

Figure 2. Driving the Clock Inputs (X1 and X2) of Am8085A.

GENERATING Am8085A WAIT STATE

The following circuit may be used to insert one WAIT state in each Am8085A machine cycle.

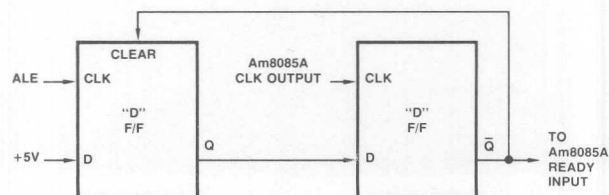


Figure 3. Generation of a Wait State for Am8085A CPU.

The D flip-flops should be chosen such that

- CLK is rising edge triggered
- CLEAR is low-level active.

BASIC SYSTEM TIMING

The Am8085A has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8 bits of address on the Data Bus. Figure 2 shows an instruction fetch, memory read and I/O write cycle (OUT). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

As in the Am9080A, the READY line is used to extend the read and write pulse lengths so that the Am8085A can be used with slow memory. Hold causes the CPU to relinquish the bus when it is through with it by floating the Address and Data Buses.

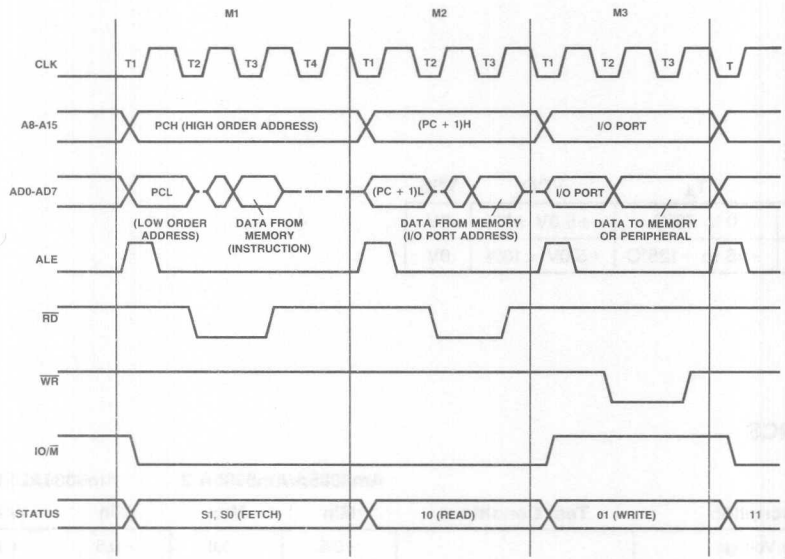


Figure 4. Am8085A Basic System Timing.

MOS-118

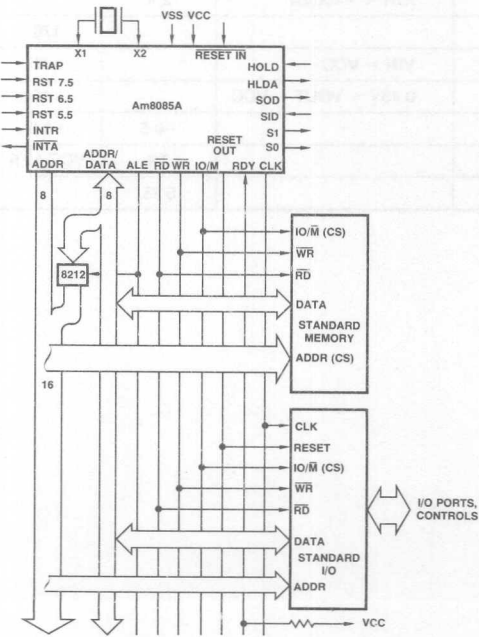


Figure 5. System Using Standard Memories.

MOS-121

Am8085A/Am8085A-2/Am9085ADM

MAXIMUM RATINGS above which useful life may be impaired

Storage Temperature	-65°C to +150°C
Ambient Temperature Under Bias	-55°C to +125°C
V _{CC} with Respect to V _{SS}	-0.5V to +7.0V
All Signal Voltages with Respect to V _{SS}	-0.5V to +7.0V
Power Dissipation	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	T _A	V _{CC}	V _{SS}
Am8085A/Am8085A-2	0 to 70°C	+5.0V ±5%	0V
Am9085ADM	-55 to +125°C	+5.0V ±10%	0V

DC CHARACTERISTICS

Parameter	Description	Test Conditions	Am8085A/Am8085A-2		Am9085ADM		Units
			Min	Max	Min	Max	
V _{IL}	Input Low Voltage		-0.5	+0.8	-0.5	+0.8	Volts
V _{IH}	Input High Voltage		2.0	V _{CC} +0.5	2.2	V _{CC} +0.5	Volts
V _{OL}	Output Low Voltage	I _{OL} = 2.0mA		0.45		0.45*	Volts
V _{OH}	Output High Voltage	I _{OH} = -400μA	2.4		2.4		Volts
I _{CC}	Power Supply Current			170		200	mA
I _{IL} †	Input Leakage	V _{IN} = V _{CC}		±10		±10	μA
I _{LO}	Output Leakage	0.45V ≤ V _{OUT} ≤ V _{CC}		±10		±10	μA
V _{ILR}	Input Low Level, RESET		-0.5	+0.8	-0.5	+0.8	Volts
V _{IHHR}	Input High Level, RESET		2.4	V _{CC} +0.5	2.4	V _{CC} +0.5	Volts
V _{HY}	Hysteresis, RESET		0.25		0.25		Volts

*I_{OL} = 1.6mA

†Except Pin 1 and Pin 2.

AC CHARACTERISTICS

Parameters		Description	Am8085A		Am8085A-2		Am9085ADM		Units
			Min	Max	Min	Max	Min	Max	
tCYC	CLK Cycle Period		320	2000	200	2000	320	2000	ns
tr, tf	CLK Rise and Fall Time			30		30		30	ns
tAL	A8-A15 Valid before Trailing Edge of ALE (Note 1)		115		50		115		ns
tACL	A0-A7 Valid to Leading Edge of Control		240		115		240		ns
tXKR	X1 Rising to CLK Rising		30	120	30	100	30	120	ns
tXKF	X1 Rising to CLK Falling		30	150	30	110	30	150	ns
t1	CLK Low Time	Standard 150pF Loading	80		40		80		ns
		Lightly Loaded (Note 8)	100			100			
t2	CLK High Time	Standard 150pF Loading	120		70		120		ns
		Lightly Loaded (Note 8)	150			150			
tALL	A0-A7 Valid to Leading Edge of Control		90		50		90		ns
tLRY	ALE to READY Stable			110		30		110	ns
tLA	Address Hold Time after ALE		100		50		100		ns
tLL	ALE Width		140		80		140		ns
tLCK	ALE Low During CLK High		100		50		100		ns
tLC	Trailing Edge of ALE to Leading Edge of Control		130		60		130		ns
tAFR	Address Float after Leading Edge of $\overline{\text{READ}}$ ($\overline{\text{INTA}}$)			0		0		0	ns
tAD	Valid Address to Valid Data In			575		350		575	ns
tRD	$\overline{\text{READ}}$ (or $\overline{\text{INTA}}$) to Valid Data			300		150		300	ns
tRDH	Data Hold Time after $\overline{\text{READ}}$ ($\overline{\text{INTA}}$) (Note 7)		0		0		0		ns
tRAE	Trailing Edge of $\overline{\text{READ}}$ to Re-Enabling of Address		150		90		150		ns
tCA	Address (A8-A15) Valid after Control		120		60		120		ns
tDW	Data Valid to Trailing Edge of $\overline{\text{WRITE}}$		420		230		420		ns
tWD	Data Valid after Trailing Edge of $\overline{\text{WRITE}}$		100		60		100		ns
tCC	Width of Control Low ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{INTA}}$)		400		230		400		ns
tCL	Trailing Edge of Control to Leading Edge of ALE		50		25		50		ns
tARY	READY Valid from Address Valid			220		100		220	ns
tRYS	READY Setup Time to Leading Edge of CLK		110		100		110		ns
tRYH	READY Hold Time		0		0		0		ns
tHACK	HLDA Valid to Trailing Edge of CLK		110		40		110		ns
tHABF	Bus Float after HLDA			210		150		210	ns
tHABE	HLDA to Bus Enable			210		150		210	ns
tLDR	ALE to Valid Data In			460		270		460	ns
tRV	Control Trailing Edge to Leading Edge of Next Control		400		220		400		ns
tAC	A8-A15 Valid to Leading Edge of Control (Note 1)		270		115		270		ns
tHDS	HOLD Setup Time to Trailing Edge of CLK		170		120		170		ns
tHDH	HOLD Hold Time		0		0		0		ns
tINS	INTR Setup Time to Falling Edge of CLK, also RST and TRAP		160		150		160		ns
tINH	INTR Hold Time		0		0		0		ns

- Notes: 1. A8-A15 Address Specs apply to $\overline{\text{IO/M}}$, S0 and S1. Except A8-A15 are undefined during T4-T6 of OF cycle whereas $\overline{\text{IO/M}}$, S0 and S1 are stable.
2. Test Conditions: tCYC = 320ns (Am8085A)/200ns (Am8085A-2); CL = 150pF.
3. For all output timing where CL = 150pF use the following correction factors.
- 25pF \leq CL < 150pF: -.10ns/pF
- 150pF < CL \leq 300pF: +.30ns/pF
4. Output timings are measured with purely capacitive load.
5. All timings are measured at output voltage VL = 0.8V, VH = 2.0V and 1.5V with 20ns rise and fall time on inputs.
6. To calculate timing specifications at other values of tCYC use the table on Page 7-21.
7. Data Hold Time is guaranteed under all loading conditions.
8. Loading equivalent to 50pF + 1 TTL input.

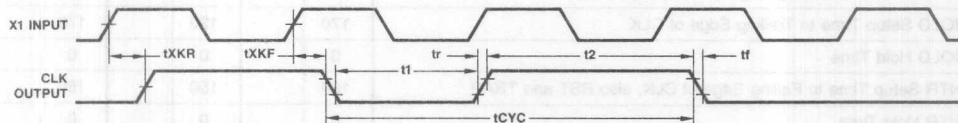
BUS TIMING SPECIFICATION AS A TCYC DEPENDENT

Parameters	Description	Am8085A/Am9085ADM		Am8085A-2		Units
		Min	Max	Min	Max	
tAL	Address Valid before Trailing Edge of ALE	$(1/2)T - 45$		$(1/2)T - 50$		ns
tLA	Address Hold Time after ALE	$(1/2)T - 60$		$(1/2)T - 50$		ns
tLL	ALE Width	$(1/2)T - 20$		$(1/2)T - 20$		ns
tLCK	ALE Low During CLK High	$(1/2)T - 60$		$(1/2)T - 50$		ns
tLC	Trailing Edge of ALE to Leading Edge of Control	$(1/2)T - 30$		$(1/2)T - 40$		ns
tAD	Valid Address to Valid Data In		$(5/2+N)T - 225$		$(5/2+N)T - 150$	ns
tRD	READ (or INTA) to Valid Data		$(3/2+N)T - 180$		$(3/2+N)T - 150$	ns
tRAE	Trailing Edge of READ to Re-Enabling of Address	$(1/2)T - 10$		$(1/2)T - 10$		ns
tCA	Address (A8-A15) Valid after Control	$(1/2)T - 40$		$(1/2)T - 40$		ns
tDW	Data Valid to Trailing Edge of WRITE	$(3/2+N)T - 60$		$(3/2+N)T - 70$		ns
tWD	Data Valid after Trailing Edge of WRITE	$(1/2)T - 60$		$(1/2)T - 40$		ns
tWDL	Leading Edge of WRITE to Data Valid		40		40	ns
tCC	Width of Control LOW (RD, WR, INTA)	$(3/2+N)T - 80$		$(3/2+N)T - 70$		ns
tCL	Trailing Edge of Control to Leading Edge of ALE	$(1/2)T - 110$		$(1/2)T - 75$		ns
tARY	READY Valid from Address Valid		$(3/2)T - 260$		$(3/2)T - 200$	ns
tHACK	HLDA Valid to Trailing Edge of CLK	$(1/2)T - 50$		$(1/2)T - 60$		ns
tHABF	Bus Float after HLDA		$(1/2)T + 50$		$(1/2)T + 50$	ns
tHABE	HLDA to Bus Enable		$(1/2)T + 50$		$(1/2)T + 50$	ns
tAC	Address Valid to Leading Edge of Control	$(2/2)T - 50$		$(2/2)T - 85$		ns
t1	CLK Low Time	$(1/2)T - 80$		$(1/2)T - 60$		ns
t2	CLK High Time	$(1/2)T - 40$		$(1/2)T - 30$		ns
tRV	Control Trailing Edge to Leading Edge of Next Control	$(3/2)T - 80$		$(3/2)T - 80$		ns
tLDR			$(4/2)T - 180$		$(4/2)T - 130$	ns
tLDW	Trailing Edge of ALE to Valid Data During WRITE		200		200	ns

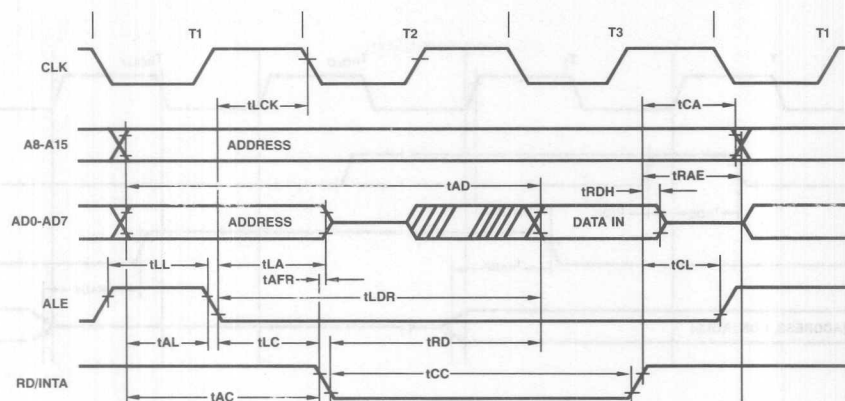
Note: N is equal to the total WAIT states.

T = tCYC.

CLOCK TIMING WAVEFORM

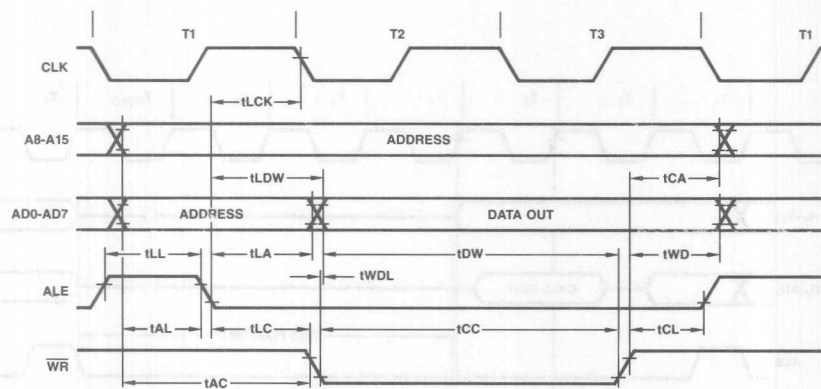


READ OPERATION



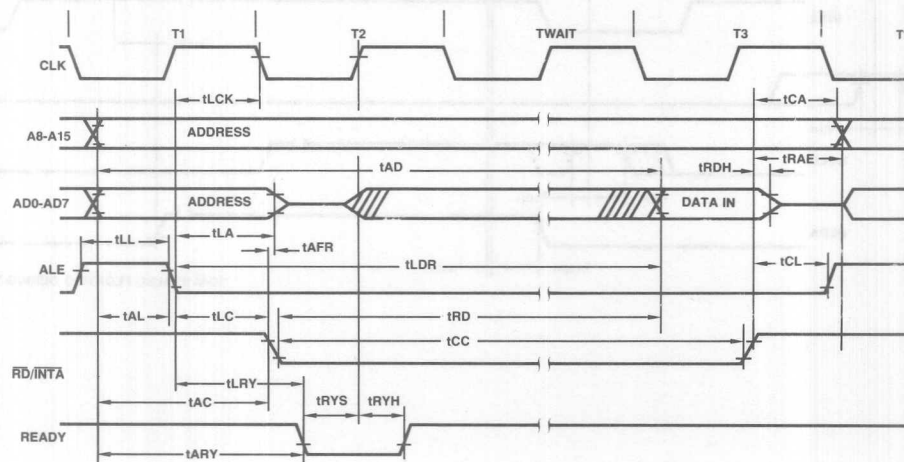
MOS-270

WRITE OPERATION



MOS-271

TYPICAL READ OPERATION WITH WAIT CYCLE



Same READY timing applies to WRITE operation.

Figure 6. Am8085A/Am8085A-2 Bus Timing

MOS-272

HOLD OPERATION

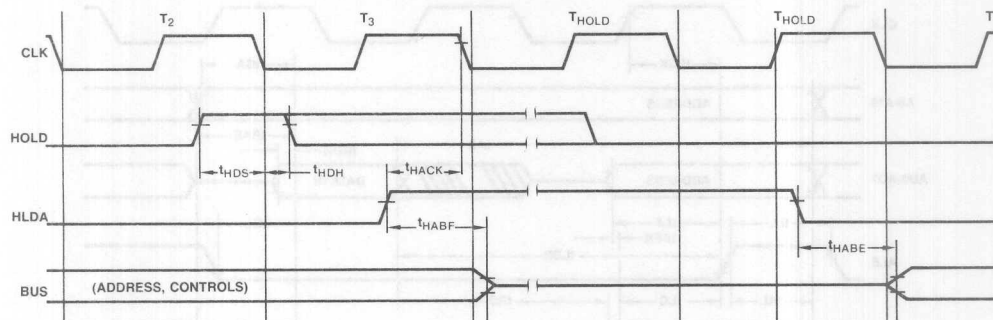
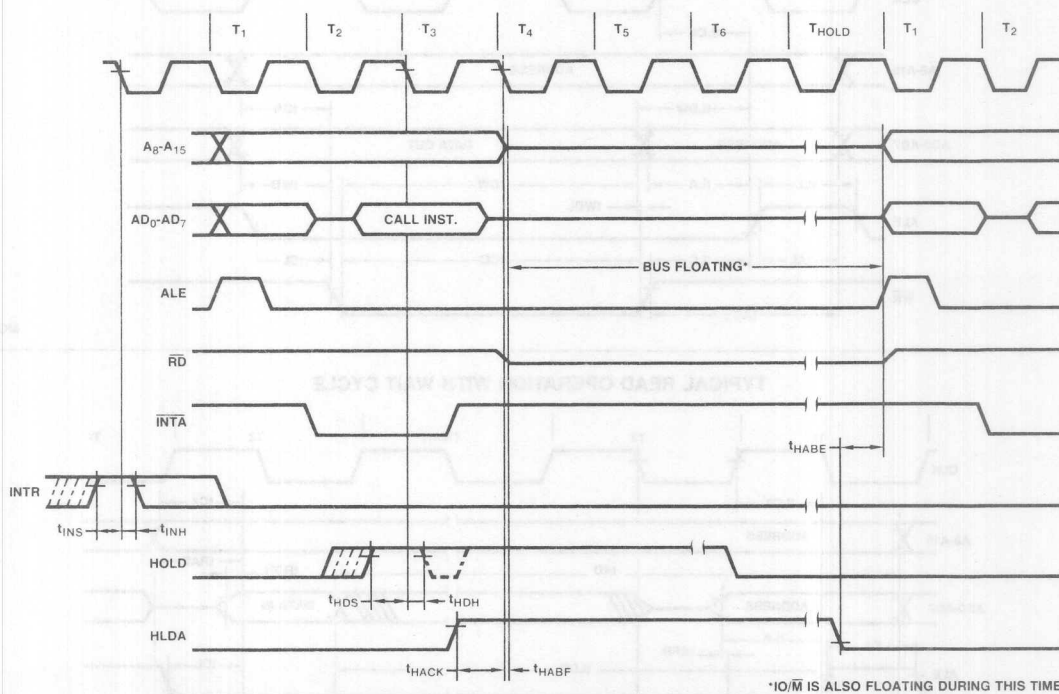


Figure 7. Am8085A Hold Timing.

MOS-130



MOS-131

Figure 8. Am8085A Interrupt and Hold Timing.

INSTRUCTION SET SUMMARY

Mnemonic*	Description	Instruction Code (Note 1)								Clock Cycles (Note 2)
		D7	D6	D5	D4	D3	D2	D1	D0	
MOVE, LOAD AND STORE										
MOVr1r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV Mr	Move register to memory	0	1	1	1	0	S	S	S	7
MOV rM	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10

INSTRUCTION SET SUMMARY (Cont.)

Mnemonic*	Description	D7	D6	D5	D4	D3	D2	D1	D0	Clock Cycles (Note 2)
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupts	1	1	1	1	0	0	1	1	4
NOP	No operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
NEW Am8085A INSTRUCTIONS										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

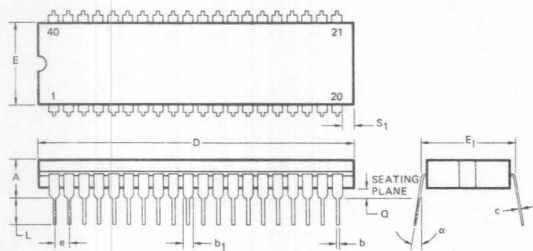
Notes: 1. DOD or SSS: 8=000, C=001, D=010, E=011, H=100, L=101, Memory=110, A=111.

2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.

*All mnemonics copyright © Intel Corporation 1977

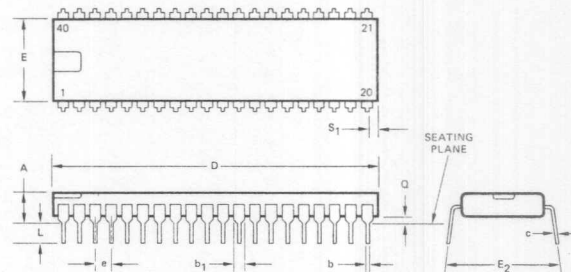
PHYSICAL DIMENSIONS Dual-In-Line

40-Pin Cerdip



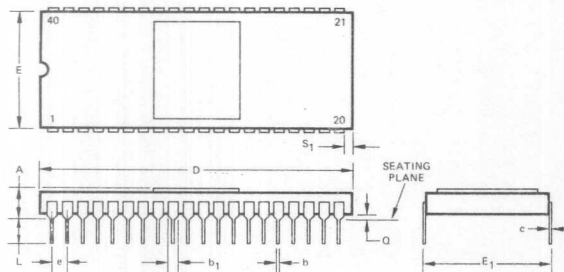
Reference Symbol	Inches	
	Min	Max
A	.150	.225
b	.016	.020
b1	.045	.065
c	.009	.011
D	2.020	2.100
E	.510	.550
E1	.600	.630
e	.090	.110
L	.120	.150
Q	.015	.060
S1	.005	
α	3°	13°

40-Pin Molded DIP



Reference Symbol	Inches	
	Min	Max
A	.150	.200
b	.015	.020
b1	.055	.065
c	.009	.011
D	2.050	2.080
E	.530	.550
E2	.585	.700
e	.090	.110
L	.125	.160
Q	.015	.060
S1	.040	.070

40-Pin Hermetic DIP



Reference Symbol	Inches	
	Min	Max
A	.100	.200
b	.015	.022
b1	.030	.060
c	.008	.013
D	1.960	2.040
E	.550	.610
E1	.590	.620
e	.090	.110
L	.120	.160
Q	.020	.060
S1	.005	

Advanced Micro Devices Commitment to Excellence

Product Assurance Programs for Military and Commercial Integrated Circuits

Advanced Micro Devices was conceived on the premise that there was a place in the semiconductor community for a manufacturer dedicated to excellence.

In product assurance procedures, Advanced Micro Devices is unique. Only Advanced Micro Devices processes all integrated circuits, commercial as well as military, to the demanding requirements of MIL-STD-883. The Rome Air Development Center (RADC), which is the Air Force's principal authority on component reliability, has issued MIL-HDBK-217B which indicates that parts processed to Military Standard 883, Level C (Advanced Micro Devices' standard processing) yield a product nearly ten times better in failure rates than the industry commercial average.

Our Sunnyvale facility has been certified by the Defense Electronics Supply Center (DESC) to produce parts to JAN Class B and C under Military Specification MIL-M-38510. The National Aeronautics and Space Administration (NASA) has certified this production line for the manufacture of Class A products for programs requiring the highest levels of reliability. Advanced Micro Devices is the only integrated circuit company formed within the last ten years to achieve such line certification.

This brochure outlines Advanced Micro Devices' standard programs for Class B, C and A devices for military and commercial operating range applications. These will cover the majority of system requirements today. Alternative screening flows for specific user needs can be performed on request. Check with your local sales office for further information.

ADVANCED MICRO DEVICES' STANDARD PRODUCTS ARE MANUFACTURED TO MIL-STD-883 REQUIREMENTS

Advanced Micro Devices' product assurance programs are based on two key documents.

MIL-M-38510 – General Specification for Microcircuits

MIL-STD-883 – Test Methods and Procedures for Microelectronics

The screening charts in this brochure show that every integrated circuit shipped by Advanced Micro Devices receives the critical screening procedures defined in MIL-STD-883, Method 5004 for Class C product. This includes molded plastic devices.

In addition, documentation, design, processing and assembly workmanship guidelines are patterned after MIL-M-38510 specifications.

Commercial and industrial users receive the quality and reliability benefits of this aerospace-type screening and documentation at no additional cost.

STANDARD PRODUCT TESTING CATEGORIES

Advanced Micro Devices offers integrated circuits to four standard testing categories.

1. Commercial operating range product (typically 0°C to 70°C)
2. Commercial product with 100% temperature testing
3. Military operating range product (typically –55°C to +125°C)
4. JAN qualified product

Categories 1, 2 and 3 are available on most Advanced Micro Devices circuits. Category 4 is offered on a more limited line. Check with your local sales office for details.

STANDARD PRODUCT ASSURANCE CATEGORIES

Devices produced to the above testing categories are available to the three standard classes of product assurance defined by MIL-STD-883. As a minimum, every device shipped by Advanced Micro Devices meets the screening requirements of Class C.

Class C – For commercial and ground-based military systems where replacement can be accomplished without difficulty.

According to MIL-HDBK-217B, this assures relative failure rates 9.4 times better than that of regular industry commercial product.

Class B – For flight applications and commercial systems where maintenance is difficult or expensive and where reliability is vital.

Devices are upgraded from Class C to Class B by burn-in screening and additional testing.

According to MIL-HDBK-217B, Class B failure rate is improved 30 times over regular industry commercial product. Advanced Micro Devices Class B processing conforms to MIL-STD-883 requirements. MIL-HDBK-217B indicates that this may provide failure rates as much as two times better than some other manufacturers' "equivalent" or "pseudo" Class B programs.

Class S – For space applications where replacement is extremely difficult or impossible and reliability is imperative.

Class S screening includes x-ray and other special inspections tailored to the specific requirements of the user.

The 100% screening and quality conformance testing performed within these Advanced Micro Devices programs is shown in TABLES I, II and III. A full description of the process flow is provided in Product Assurance Document 15-010, available on request.

CLASS C SCREENING FLOW FOR COMMERCIAL SYSTEMS AND GROUND BASED MILITARY SYSTEMS

TABLE I
CLASS C
INTEGRATED CIRCUITS

Screening Procedure per MIL-STD-883 Method 5004, Class C		COMMERCIAL OPERATING RANGE	MILITARY OPERATING RANGE	
		HERMETIC AND MOLDED PACKAGES	HERMETIC PACKAGE ONLY	
Screen	Test Method	Flow C1 Commercial Product	Flow C3 Military Product	Flow C4 Jan Qualified Product
VISUAL AND MECHANICAL				
Internal visual	2010, Condition B	100%	100%	100%
High temperature storage	1008, Condition C, 24 hours	100%	100%	100%
Temperature cycle	1010, Condition C	100%	100%	100%
Constant acceleration	2001	100% (1)	100%	100%
Hermeticity, Fine and Gross	1014	100% (1)	100%	100%
FINAL ELECTRICAL TESTS		AMD Data Sheet	AMD Data Sheet	38510 Slash Sheet
Static (dc)	a) At 25°C, and power supply extremes b) At temperature and power supply extremes	100% (2)	100% —	100% —
Functional	a) At 25°C, and power supply extremes b) At temperature and power supply extremes	100% (2)	100% —	100% —
Switching (ac) or Dynamic	At 25°C, nominal power supply	(2)	—	—
QUALITY CONFORMANCE	5005, Group A (See Table II)	Sample	Sample	Sample
Sample Tests	Group B Group C Group D	— — —	— — —	Sample Sample Sample
EXTERNAL VISUAL	2009 (Note 5)	100%	100%	100%

TABLE II
GROUP A QUALITY CONFORMANCE LEVELS

Advanced Micro Devices employs the military-recommended LTPD sampling system to assure quality. MIL-STD-883, Method 5005, TABLE I, Group A, subgroups 1 through 9 as appropriate to the device family are performed on every lot. Quality levels defined for Class B product are applied by Advanced Micro Devices to both Class B and Class C orders.

	LTPD	INITIAL SAMPLE SIZE
Subgroup 1 – Static tests at 25°C	5	45
Subgroup 2 – Static tests at maximum rated operating temperature	7	32
Subgroup 3 – Static tests at minimum rated operating temperature	7	32
Subgroup 4 – Dynamic tests at 25°C – LINEAR devices	5	45
Subgroup 5 – Dynamic tests at maximum rated operating temperature – LINEAR devices	7	32
Subgroup 6 – Dynamic tests at minimum rated operating temperature – LINEAR devices	7	32
Subgroup 7 – Functional tests at 25°C	5	45
Subgroup 8 – Functional tests at maximum and minimum rated operating temperatures	10	22
Subgroup 9 – Switching tests at 25°C – DIGITAL devices	7	32
Subgroup 10 – Switching tests at maximum rated operating temperatures – DIGITAL devices	*	
Subgroup 11 – Switching tests at minimum rated operating temperatures – DIGITAL devices	*	

*These subgroups, where applicable, are usually performed during initial characterization only for all except JAN Qualified product.

CLASS B SCREENING FLOW FOR HIGH RELIABILITY COMMERCIAL AND MILITARY SYSTEMS

TABLE III
CLASS B
INTEGRATED CIRCUITS
(Class C plus burn in screening
and additional testing.)

TABLE III CLASS B INTEGRATED CIRCUITS (Class C plus burn in screening and additional testing.)		COMMERCIAL OPERATING RANGE	MILITARY OPERATING RANGE	
		HERMETIC AND MOLDED PACKAGES	HERMETIC PACKAGE ONLY	
		Flow B1	Flow B3	Flow B4
Screening Procedure per MIL-STD-883 Method 5004, Class B		Commercial Product	Military Product	Jan Qualified Product
Screen	Test Method			
VISUAL AND MECHANICAL				
Internal visual	2010, Condition B	100%	100%	100%
High temperature storage	1008, Condition C, 24 hours	100%	100%	100%
Temperature cycle	1010, Condition C	100%	100%	100%
Constant acceleration	2001	100% (1)	100%	100%
Hermeticity, Fine and Gross	1014	100% (1)	100%	100%
BURN IN				
Interim (pre burn in) electricals	Per applicable device specification	100%	100%	100%
Burn in	1015, 160 hours at 125°C or equivalent.*	100% (3)	100%	100%
FINAL ELECTRICAL TESTS		AMD Data Sheet	AMD Data Sheet	38510 Slash Sheet
Static (dc)	a) At 25°C, and power supply extremes	100%	100%	100%
Functional	b) At temperature and power supply extremes	(2) (3)	100%	100%
	a) At 25°C, and power supply extremes	100%	100%	100%
	b) At temperature and power supply extremes	(2) (3)	100%	100%
Switching (ac) or Dynamic	At 25°C, nominal power supply	(2)	100%	100%
QUALITY CONFORMANCE		Sample	Sample	Sample
Sample Tests	5005, Group A (See Table II)	—	(4)	Sample
	Group B	—	(4)	Sample
	Group C	—	(4)	Sample
	Group D	—	(4)	Sample
EXTERNAL VISUAL	2009 (Note 5)	100%	100%	100%

Notes: 1. Not applicable to molded packages.

2. All MOS RAMs and many other MOS devices receive a.c. testing and 100% d.c. screening at high temperature and power supply extremes as standard. Other products sampled at Group A (Table II).

3. Am2900 LSI products receive a 96 hour burn-in, plus 100% d.c. screening at high temperature and power supply extremes.

4. Available to special order.

5. Without optical aid for commercial devices.

*(Unless device data sheet specifies otherwise).

CLASS S FOR AEROSPACE SYSTEMS. (FORMERLY CLASS A)

Advanced Micro Devices offers Class S programs based on screening defined in MIL-STD-883, Method 5004.

Contact your local Advanced Micro Devices' sales office for more information.

STANDARD PRODUCT SCREENING SUMMARY AND ORDERING INFORMATION

1. COMMERCIAL PRODUCT

- Screened per MIL-STD-883, Method 5004.
- Electrically tested per AMD Data Sheet.
- Supplied in hermetic and molded packages.
- Quality conformance testing, Method 5005, Group A, performed to levels specified for Class B on both Class C and Class B options.

Class C (Flow C1)

- Order standard AMD part number.
- Marked same as order number.

Example: Am2901ADC

Class B (Flow B1)

- Burn in performed in AMD circuit condition.
- Order standard AMD part number, add suffix B (or /883B for 1, 2 and 300 Series Linear devices).
- Marked same as order number.

Example: Am2901ADC-B

3. MILITARY PRODUCT

- Screened per MIL-STD-883, Method 5004.
- Electrically tested per AMD Data Sheet.
- Supplied in hermetic package only.
- Quality conformance testing, Method 5005, Group A, performed to levels specified for Class B on both Class B and Class C options.

Class C (Flow C3)

- Order standard AMD part number.
- Marked same as order number.

Example: Am2901ADM

Class B (Flow B3)

- Burn in performed in AMD circuit condition.
- AC at 25°C, dc and functional testing at 25°C as well as temperature and power supply extremes performed on 100% of every lot.
- Quality conformance testing, Method 5005, Groups B, C and D available to special order.
- Order standard AMD part number, add suffix B.
- Marked same as order number.

Example: Am2901ADM-B

2. COMMERCIAL PRODUCT WITH 100% TEMPERATURE TESTING

- Identical to standard commercial operating range product with the addition of 100% dc and functional testing at 100°C and power supply extremes.

Class C (Flow C2)

- Order standard AMD part number, add suffix T.
- Marked same as order number.

Example: Am2901ADC-T

Class B (Flow B2)

- Burn in performed in AMD circuit condition.
- Order standard AMD part number, add suffix TB.
- Marked same as order number.

Example: Am2901ADC-TB

4. JAN QUALIFIED PRODUCT

- Screened per MIL-STD-883, Method 5004.
- Electrically tested to JAN detail specification (slash sheet).
- Manufactured in Defense Logistics Agency certified facility.
- Quality conformance testing, Method 5005, Groups A, B, C and D performed as standard and must be completed prior to shipment.
- It is a product for which AMD has gained QPL listing.*

Class C (Flow C4)

- Order per military document.
- Marked per military document.

Example: JM38510/44001CQB

Class B (Flow B4)

- Burn in performed in circuit condition approved for JAN devices.
- Order per military document.
- Marked per military document.

Example: JM38510/44001BRC

*In certain cases where JAN Qualified product is specified but is not available, Advanced Micro Devices can provide devices to the electrical limits and burn-in criteria of the slash sheet. This class of product has been called JAN Equivalent and marked M38510/ by some manufacturers. This identification is no longer permitted by DESC. Check with your local sales office for availability of specific device types.

SALES OFFICES AND REPRESENTATIVES

SOUTHWEST AREA

Advanced Micro Devices
9595 Wishire Boulevard
Suite 401
Beverly Hills, California 90212
Tel: (213) 278-9700
(213) 278-9701
TWX: 910-490-2143

Advanced Micro Devices
1414 West Broadway Road
Suite 239
Tempe, Arizona 85282
Tel: (602) 244-9511
TWX: 910-950-0127

Advanced Micro Devices
4000 MacArthur Blvd.
Suite 5000
Newport Beach, California 92660
Tel: (714) 752-6262
TWX: 910-585-1575

Advanced Micro Devices
13771 No. Central Expy.
Suite 1008
Dallas, Texas 75243
Tel: (214) 234-5886
TWX: 910-867-4795

Advanced Micro Devices
5955 Desoto Ave., Suite 249
Woodland Hills, California 91367
Tel: (213) 992-4155
TWX: 910-494-4720

Advanced Micro Devices
9455 Ridgehaven Court
Suite 230
San Diego, CA 92123
Tel: (714) 560-7030

NORTHWEST AREA

Advanced Micro Devices
3350 Scott Boulevard
Suite 1002, Bldg. 10
Santa Clara, California 95051
Tel: (408) 727-1300
TWX: 910-538-0182

Advanced Micro Devices
7000 Broadway
Suite 401
Denver, Colorado 80221
Tel: (303) 426-7100
TWX: 910-931-2562

Advanced Micro Devices
7110 S.W. Fir Loop, Suite 130
Tigard, Oregon 97223
Tel: (503) 620-1021
TWX: 910-458-8797

Advanced Micro Devices
Honeywell Ctr., Suite 1002
600 108th Ave. N.E.
Bellevue, Washington 98004
Tel: (206) 455-3600

MID-AMERICA AREA

Advanced Micro Devices
1111 Plaza Drive, Suite 420
Schaumburg, Illinois 60195
Tel: (312) 882-8660
TWX: 510-291-3589

Advanced Micro Devices
3400 West 66th Street
Suite 375
Edina, Minnesota 55435
Tel: (612) 929-5400
TWX: 910-576-0929

Advanced Micro Devices
50 McNaughton Road
Suite 102
Columbus, Ohio 43213
Tel: (614) 864-9906
TWX: 810-339-2431

Advanced Micro Devices
33150 Schoolcraft, Suite 104
Livonia, Michigan 48150
Tel: (313) 425-3440
TWX: 810-242-8777

MID-ATLANTIC AREA

Advanced Micro Devices
40 Crossway Park Way
Woodbury, New York 11797
Tel: (516) 364-8020
TWX: 510-221-1819

Advanced Micro Devices
6806 Newbrook Ave
E. Syracuse, New York 13057
Tel: (315) 437-7546
TELEX: 93-7201

Advanced Micro Devices
2 Kilmer Road
Edison, New Jersey 08817
Tel: (201) 985-6800
TWX: 710-480-6260

Advanced Micro Devices
1 Gibraltar Plaza, Suite 219
Horsesham, Pennsylvania 19044
Tel: (215) 441-8210
TWX: 510-565-7572

Advanced Micro Devices
82 Washington Street, Suite 206
Poughkeepsie, New York 12601
Tel: (914) 471-8180
TWX: 510-248-4219

NORTHEAST AREA

Advanced Micro Devices
300 New Boston Park
Woburn, Massachusetts 01801
Tel: (617) 933-1234
TWX: 710-348-1332

SOUTHEAST AREA

Advanced Micro Devices
793 Elbridge Landing, #11N
Linthicum, Maryland 21090
Tel: (301) 736-9310
TWX: 710-861-0503

Advanced Micro Devices
1001 N.W. 62nd Street
Suite 100
Ft. Lauderdale, Florida 33309
Tel: (305) 771-6510
TWX: 510-955-9490

Advanced Micro Devices
6755 Peachtree Industrial Boulevard
Suite 104
Atlanta, Georgia 30360
Tel: (404) 445-7920
TWX: 810-769-0430

Advanced Micro Devices
501 Archdale Dr., Suite 227
Charlotte, North Carolina 28210
Tel: (704) 525-1875

Advanced Micro Devices International Sales Offices

BELGIUM

Advanced Micro Devices
Overseas Corporation
Avenue de Tenueren, 412, bte 9
B-1150 Bruxelles
Tel: (02) 771 99 93
TELEX: 61028

FRANCE

Advanced Micro Devices, S.A.
European Marketing Centre
27, Blvd. Général-Vautrin
F-92400 Cannes
Tel: (093) 73.60.75
TELEX: 470966

Advanced Micro Devices, S.A.
Silec 314, Immeuble Heisink
74, rue d'Arcueil
F-94588 Rungis Cedex
Tel: (01) 686.91.86
TELEX: 202053

GERMANY

Advanced Micro Devices GmbH
Rosenheimer Str. 139
D-8000 München 83
Tel: (089) 514 0722
TELEX: 0-523883

Advanced Micro Devices GmbH
Herthauser Hauptstrasse 4
D-7024 Filderstadt
Tel: (07158) 6 26 3 0
TELEX: 0-721211

ITALY

Advanced Micro Devices S.R.L.
Centro Direzionale
Palazzo Vasari, 3° Piano
I-20090 Mi2 - Segrate (MI)
Tel: (02) 215 4913-4-5
TELEX: 315286

JAPAN

Advanced Micro Devices, K.K.
Dai 3 Hoya Building
8-17, Kamiakado 1-chome
Suginami-ku, Tokyo 168
Tel: (03) 329-2751
TELEX: 2324064

Advanced Micro Devices, K.K.
Daichō-Seimek Ezaka Dai-2 Bldg.
23-5, 1-chome, Ezaka-cho, Suita-shi
Osaka 564
Tel: (06) 386-9161

SWEDEN

Advanced Micro Devices AB
Box 7013
S-172 07 Sundbyberg
Tel: (08) 96 12 35
TELEX: 11602

UNITED KINGDOM

Advanced Micro Devices (U.K.) Ltd.
A.M.D. House,
Goldsworth Road,
Woking,
Surrey GU21 1JT
Tel: Woking (04862) 22121
TELEX: 859103

International Sales Representatives and Distributors

ARGENTINA

Thiko, S.A. Electronica
Peru 653
1068 - Buenos Aires
Argentina
Tel: (1) 30-4132
(1) 33-4870
TELEX: 17825
Kof International Corporation
1660 Rollins Road
Burlingame, California 94010
Tel: (415) 692-3200
TELEX: 331491

AUSTRALIA

A.I. Distributors Pty. Ltd.
P.O. Box 71
44 Prospect Road
Prospect, S.A. 5082
Tel: (8) 269-1244
TELEX: 82635

R and D Electronics
257 Burwood Highway
P.O. Box 206
Burwood 3125
Victoria
Tel: (03) 288-8232/82
TELEX: AA33288

R and D Electronics
P.O. Box 57
Crows Nest N.S.W. 2065
Sydney
Tel: (61) 439-5488
TELEX: (790) 25468

AUSTRIA

Kontron Ges.m.b.H.
Industriest. B 13
A-2345 Brunn am Gebirge
Tel: (02236) 8 66 31
TELEX: 79337

BELGIUM

MCA Tronix S.P.R.L.
Route du Condroz, 513
B-4200 Ougree
Tel: (041) 362780/362795
TELEX: 42052

DENMARK

Advanced Electronic of Denmark Aps
Mariendalsvej 55
DK-2000 Copenhagen F
Tel: (01) 19 44 33
TELEX: 22 431

FINLAND

Komdel OY
Vanha Finncentie 4
Box 32
SF-02271 Espoo 27
Tel: (01) 885 011
TELEX: 12 1926

FRANCE

A2M
6, Avenue du Général De Gaulle
Hall A
F-78150 Le Chesnay
Tel: (01) 954.91.13
TELEX: 698 376

Lyon-Electronic-Direlec (LED)
18, rue Henri Poincaré
F-69352 Lyon Cedex 2
Tel: (078) 58.85.85
TELEX: 300 270

RTF

73, avenue Charles De Gaulle
F-92200 Neuilly-sur-Seine
Tel: (01) 747.11.01
TELEX: 611985

GERMANY

Cosmos Elektronik GmbH
Hegelestrasse 16
D-6000 München 83
Tel: (089) 60 20 88
TELEX: 0-522545
EBV-Elektronik Vertriebs GmbH
Obenweg 6
D-8025 Unterhaching b. München
Tel: (089) 6 11 05-1
TELEX: 0-524535
EBV-Elektronik Vertriebs GmbH
Oststr. 129
D-4000 Düsseldorf
Tel: (0211) 8 48 46/7
TELEX: 0-5887267

EBV-Elektronik Vertriebs GmbH
Kreibitzrain 18
D-3006 Burgwedel 1/Hannover
Tel: (05139) 50 38
TELEX: 0-923694
EBV-Elektronik Vertriebs GmbH
Myliusstr. 54
D-6000 Frankfurt 1
Tel: (0611) 72 04 16
TELEX: 0-413590

EBV-Elektronik Vertriebs GmbH
Alexandersstr. 42
D-7000 Stuttgart 1
Tel: (0714) 24 78 1/83
TELEX: 0-722271
Elbalex GmbH
Caselienstr. 24
D-7100 Heilbronn
Tel: (07131) 89 00 1
TELEX: 0-728362

Nordelektronik Vertriebs GmbH
Bahnhofstr. 14
D-2301 Kiel-Raisdorf
Tel: (04307) 54 83
Nordelektronik Vertriebs GmbH
Harkshedenweg 238-240
D-2085 Quickborn
Tel: (04106) 40 31
TELEX: 0-214299

MEV-Mikro Elektronik Vertrieb
H. Lichotka GmbH
Muenchener Strasse 16A
D-8021 Dinning b. München
Tel: (08170) 71 61
TELEX: 0-527828

INDIA
SRI RAM Associates
P.O. Box 60965
Sunnyvale, CA 94088
Tel: (408) 738-2295
TELEX: 346369
SRI RAM Associates
245 Jayanagara III Block
Bangalore 560011
Tel: 611606
TELEX: 845-8162

ISRAEL
Talviton Electronics, Ltd.
P.O. Box 21104
9 Bitmote Street
Tel-Aviv
Tel: (03) 444572
TELEX: 03-3400

ITALY
Indelco, S.R.L. - Milano
Via S. Sempliciano, 2
I-20121 Milano
Tel: (02) 862963

Indelco, S.R.L. - Roma
Via C. Colombo, 134
I-00147 Roma
Tel: (06) 514 0722
TELEX: 611517
Cramer Italia S.p.A.
Via Ferrarese, 10/2
I-40128 Bologna
Tel: (051) 372777
TELEX: 511870

Cramer Italia S.p.A.
Via S. Sempliciano, 2
I-20121 Milano
Tel: (02) 809326
Cramer Italia S.p.A.
3° Traversa Domenico Fontana, 22 A/B
I-80128 Napoli
Tel: (081) 255300

Cramer Italia S.p.A.
Via C. Colombo, 134
I-00147 Roma
Tel: (06) 517981
TELEX: 611517
Cramer Italia S.p.A.
Corso Trastevere, 109
I-10127 Torino
Tel: (011) 619 2062/2067
TELEX: 211252

JAPAN
Advanced Technology Corporation
of Japan
Tashi Bldg., 3rd Floor
No. 8, Minami Motomachi
Shinjuku-ku, Tokyo 160
Tel: (03) 265-9416

Asahi Glass Company, Ltd.
2-1-2, Marunouchi
Chiyoda-ku, Tokyo 100
Tel: (03) 218-5680
Dainichi Electronics
Kohraku Building
1-8, 1-chome, Kohraku
Bunkyo-ku, Tokyo
Tel: (03) 819-6876

Dainichi Electronics
Kintotsu-Takama Building
38-3 Takama-cho
Narashi 630
ISI Ltd.
B-3, 4-chome, Udagashi
Chiyoda-ku, Tokyo 102
Tel: (03) 264-3301

Kanematsu Denchi K.K.
Takanawa Bldg., 2nd Floor
19-25, 3-chome, Takanawa
Minato-ku, Tokyo 108
Microtek, Inc.
Nato Building
7-2-8 Nishishinjuku
Shinjuku-ku, Tokyo 160
Tel: (03) 363-2317
TELEX: J28497

KOREA
Dukung Trading Co.
Room 301 - Jinwon Bldg. 507-30
Sinim 4-Dong
Gwanak-ku
Seoul
Tel: 856-9764
TELEX: K23459

NETHERLANDS

Arcofel BV
Van Almondestraat 6
P.O. Box 344
NL-5340 AE Oss
Tel: (04120) 24200/27574
TELEX: 50835

NEW ZEALAND

G.T.S. Engineering, Ltd.
P.O. Box 42-053, Orakei
Auckland
Tel: (Auck) 543 214
(Auck) 546 745
TELEX: NZ2591

NORWAY

A.S. Kjell Bakke
Nygaata 48
P.O. Box 143
N-2011 Strømmen
Tel: (02) 71 53 50
TELEX: 19407

SINGAPORE

Sentry-Tech (Pte.) Ltd.
3210 Shaw Towers
100 Beach Road
Singapore 0718
Tel: 292 4342
745 4260
TELEX: RS 34235

SOUTH AFRICA

South Continental Devices (Pty.) Ltd.
P.O. Box 56420
Pinetown 2123
Tel: (011) 789-2400
TELEX: 4-24849 SA

SOUTH AMERICA

Intecra
2349 Charleston Road
Mountain View, CA 94043
Tel: (415) 967-8818/25
TELEX: 345 545

SPAIN

Regula S.A.
Avda. Baro 19
Barcelona
Tel: (03) 386 19 58
Regula S.A.
Avda. de Ramón y Cajal, 5
Madrid-16
Tel: (01) 459 33 00/04/08
TELEX: 42 207

Sagtron, S.A.
General de Importaciones Electronicas
C/ Castelló, 25, 2°
Madrid-1
Tel: 856-9764
TELEX: 43819

SWEDEN

Svensk Telexindustri AB
Box 5024
S-162 05 Vallby
Tel: (08) 89 04 35
TELEX: 13033

SWITZERLAND

Kurt Hirt AG
Thurgauerstr. 74
CH-8350 Zuerich
Tel: (01) 302 21 21
TELEX: 53461

TAIWAN

Everdata Pacific Co.
3421 Geary Blvd.
San Francisco, CA 94118
Tel: (415) 688-7524
TELEX: 171470

Everdata International Corp.
11/F 219, Chung Hsiao E. Rd.
Sec. 4, Taipei
Tel: (0222) 54051
TELEX: 21528

UNITED KINGDOM

Candy Electronic Components
52 College Road
Maldenstone, Kent ME15 6SA
Tel: (0622) 54051
TELEX: 965998

Hawke-Cramer
Hawke Electronics Limited
Amotex House
45 Hanworth Road
Sunbury-on-Thames
Middlesex TW16 5DA
Tel: (01) 879-7799
TELEX: 925892

Dage Eurosem, Ltd.
Haywood House
84 High Street
Pinner, Middlesex, HA5 5QA
Tel: (01) 868 0028/9
TELEX: 24506

ITT Electronic Services
Edinburgh Way
Harlow, Essex CM20 2DF
Tel: Harlow (0279) 26777
TELEX: 81525

Memelec Ltd.
Thame Park Industrial Estate
Thame
Oxon OX9 3RS
Tel: Thame (084 421) 3146
TELEX: 837508

Quamdon Electronics
(Semiconductors) Ltd.
Stack Lane
Derby DE3 3ED
Tel: Derby (0332) 363291
TELEX: 37163